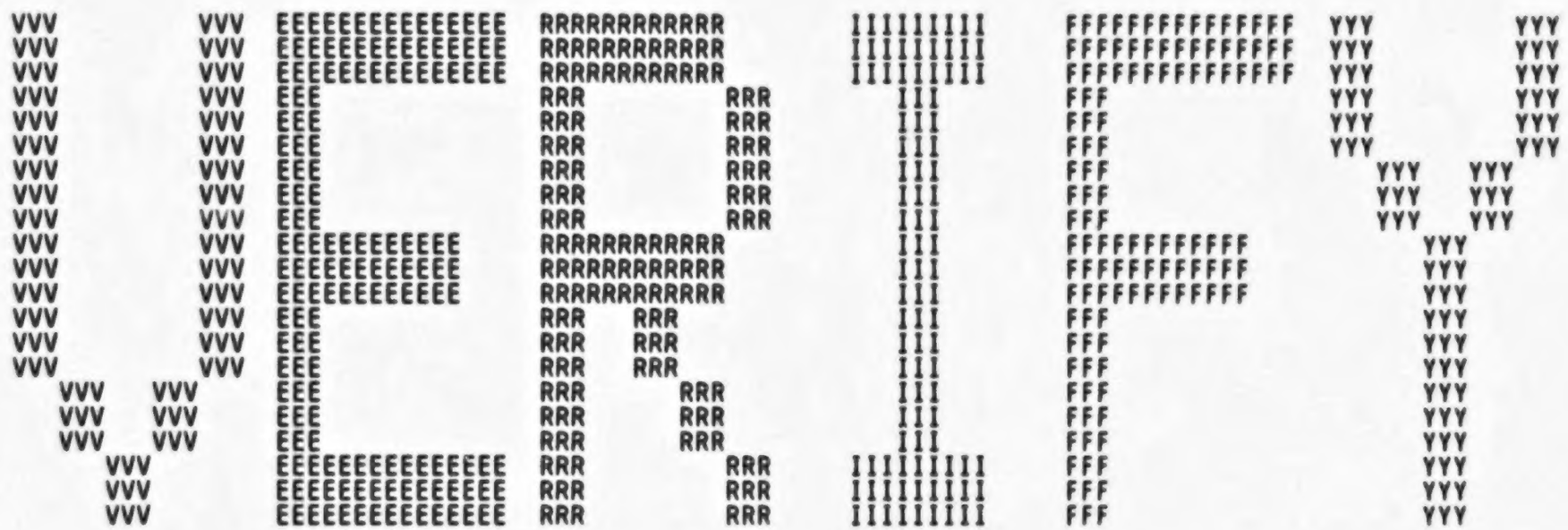
VVV VVV EEEEEEEEEEEEEEE RRRRRRRRRRR IIIIIIIII FFFFFFFFFFFFFFF YYY YYY
VVV VVV EEEEEEEEEEEEEEE RRRRRRRRRRR IIIIIIIII FFFFFFFFFFFFFFF YYY YYY
VVV VVV EEEEEEEEEEEEEEE RRRRRRRRRRR IIIIIIIII FFFFFFFFFFFFFFF YYY YYY
VVV VVV EEE RRR RRR III FFF YYY YYY
VVV VVV EEE RRR RRR III FFF YYY YYY
VVV VVV EEE RRR RRR III FFF YYY YYY
VVV VVV EEE RRR RRR III FFF YYY YYY
VVV VVV EEE RRR RRR III FFF YYY YYY
VVV VVV EEE RRR RRR III FFF YYY YYY
VVV VVV EEEEEEEEEEEEE RRRRRRRRRRR III FFFFFFFFFFFF YYY
VVV VVV EEEEEEEEEEEEE RRRRRRRRRRR III FFFFFFFFFFFF YYY
VVV VVV EEEEEEEEEEEEE RRRRRRRRRRR III FFFFFFFFFFFF YYY
VVV VVV EEE RRR RRR III FFF YYY
VVV VVV EEE RRR RRR III FFF YYY
VVV VVV EEE RRR RRR III FFF YYY
VVV VVV EEE RRR RRR III FFF YYY
VVV VVV EEE RRR RRR III FFF YYY
VVV VVV EEE RRR RRR III FFF YYY
VVV EEEEEEEEEEEEEEE RRR RRR IIIIIIIII FFF YYY
VVV EEEEEEEEEEEEEEE RRR RRR IIIIIIIII FFF YYY
VVV EEEEEEEEEEEEEEE RRR RRR IIIIIIIII FFF YYY

FILEID**VERIFY

```
VV      VV  EEEEEEEEEE  RRRRRRRR     IIIIII    FFFFFFFFFF  YY      YY
VV      VV  EEEEEEEEEE  RRRRRRRR     IIIIII    FFFFFFFFFF  YY      YY
VV      VV  EE          RR      RR     II      FF          YY      YY
VV      VV  EE          RR      RR     II      FF          YY      YY
VV      VV  EE          RR      RR     II      FF            YY  YY
VV      VV  EE          RR      RR     II      FF            YY  YY
VV      VV  EEEEEEEE    RRRRRRRR       II      FFFFFFFF        YY
VV      VV  EEEEEEEE    RRRRRRRR       II      FFFFFFFF        YY
VV      VV  EE          RR  RR         II      FF              YY
VV      VV  EE          RR  RR         II      FF              YY
  VV  VV    EE          RR    RR       II      FF              YY          ....
  VV  VV    EE          RR    RR       II      FF              YY          ....
    VV      EEEEEEEEEE  RR      RR   IIIIII    FF              YY          ....
    VV      EEEEEEEEEE  RR      RR   IIIIII    FF              YY          ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
;     1          0001  0
;     2          0002  0 MODULE VERIFY   (%TITLE 'Main module'
;     3          0003  0                 MAIN = VERIFY,
;     4          0004  0                 IDENT = 'V04-000'
;     5          0005  0                 ) =
;     6          0006  1 BEGIN
;     7          0007  1
;     8          0008  1 !
;     9          0009  1 !*****************************************************************************
;    10          0010  1 !*                                                                           *
;    11          0011  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
;    12          0012  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
;    13          0013  1 !*  ALL RIGHTS RESERVED.                                                      *
;    14          0014  1 !*                                                                           *
;    15          0015  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
;    16          0016  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
;    17          0017  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
;    18          0018  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
;    19          0019  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
;    20          0020  1 !*  TRANSFERRED.                                                              *
;    21          0021  1 !*                                                                           *
;    22          0022  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
;    23          0023  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT     *
;    24          0024  1 !*  CORPORATION.                                                              *
;    25          0025  1 !*                                                                           *
;    26          0026  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
;    27          0027  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                    *
;    28          0028  1 !*                                                                           *
;    29          0029  1 !*                                                                           *
;    30          0030  1 !*****************************************************************************
;    31          0031  1 !
;    32          0032  1
;    33          0033  1 !++
;    34          0034  1 ! FACILITY:
;    35          0035  1 !       Files-11 structure verification utility.
;    36          0036  1 !
;    37          0037  1 ! ABSTRACT:
;    38          0038  1 !       This is the main module.
;    39          0039  1 !
;    40          0040  1 ! ENVIRONMENT:
;    41          0041  1 !       VAX/VMS user mode.
;    42          0042  1 !--
;    43          0043  1 !
;    44          0044  1 ! AUTHOR: M. Jack, CREATION DATE: 11-Oct-1980
;    45          0045  1 !
;    46          0046  1 ! MODIFIED BY:
;    47          0047  1 !
;    48          0048  1 !       V03-010 BLS0328         Benn Schreiber  30-JUN-1984
;    49          0049  1 !               Ensure ctrl/y re-enabled in exit handler.
;    50          0050  1 !
;    51          0051  1 !       V03-009 BLS0273         Benn Schreiber  20-FEB-1984
;    52          0052  1 !               Add ctrl/c, ctrl/y handler to unlock volume and exit
;    53          0053  1 !               verify.
;    54          0054  1 !
;    55          0055  1 !       V03-008 MLJ0116         Martin L. Jack, 13-Aug-1983  15:50
;    56          0056  1 !               In HEADER_ERROR, cover an error for a file sequence number of
;    57          0057  1 !               -1.  This is the flag for previously deleted headers.  Update
```

```
;    58          0058  1 !                 for long filenames and long UICs.
;    59          0059  1 !
;    60          0060  1 !       V03-007 MLJ0107         Martin L. Jack, 4-Apr-1983  13:19
;    61          0061  1 !                 Update for change to FH2$C_LENGTH.
;    62          0062  1 !
;    63          0063  1 !       V03-006 MLJ0095         Martin L. Jack, 17-Aug-1982  18:29
;    64          0064  1 !                 Remove references to CLI$END_PARSE.
;    65          0065  1 !
;    66          0066  1 !       V03-005 MLJ0092         Martin L. Jack, 6-Jul-1982  13:41
;    67          0067  1 !                 Avoid generating a leading space in ODS-1 format date.
;    68          0068  1 !
;    69          0069  1 !       V03-004 MLJ0090         Martin L. Jack, 25-May-1982  16:09
;    70          0070  1 !                 Fix access violation after failure of CH$FIND_CH.
;    71          0071  1 !
;    72          0072  1 !       V03-003 MLJ0087         Martin L. Jack, 7-Apr-1982  15:00
;    73          0073  1 !                 Fix access violation in traversing work list.  Support ODS-1
;    74          0074  1 !                 revision 2 disks.
;    75          0075  1 !
;    76          0076  1 !       V03-002 MLJ0084         Martin L. Jack, 25-Mar-1982  21:32
;    77          0077  1 !                 Use a random sequence number rather than the next sequential
;    78          0078  1 !                 sequence number when rewriting a bad header with a deleted
;    79          0079  1 !                 header.  This tracks a policy change in the ACP.
;    80          0080  1 !
;    81          0081  1 !       V03-001 MLJ0083         Martin L. Jack, 22-Mar-1982  9:35
;    82          0082  1 !                 Add check for zero-length ODS-2 directory and return "directory
;    83          0083  1 !                 file has invalid format" error.
;    84          0084  1 !
;    85          0085  1 !       V02-006 MLJ0075         Martin L. Jack, 29-Jan-1982  12:39
;    86          0086  1 !                 Use FIB$V_NORECORD.
;    87          0087  1 !
;    88          0088  1 !       V02-005 MLJ0074         Martin L. Jack, 23-Jan-1982  23:24
;    89          0089  1 !                 Rename [001003] to [SYSLOST] to remove VMS dependence on
;    90          0090  1 !                 numbered directories.
;    91          0091  1 !
;    92          0092  1 !       V02-004 MLJ0061         Martin L. Jack, 30-Nov-1981  10:31
;    93          0093  1 !                 Account for $GETDVI interface change.
;    94          0094  1 !
;    95          0095  1 !       V02-003 MLJ0055         Martin L. Jack, 15-Oct-1981  20:28
;    96          0096  1 !                 Issue message if lost files not entered.  Add /USAGE qualifier
;    97          0097  1 !                 to generate disk-accounting data file.  Issue message if
;    98          0098  1 !                 directory file not named '.DIR;1'.  Implement complete storage
;    99          0099  1 !                 control block validation and reconstruction.  Use $GETDVI.  Add
;   100          0100  1 !                 checks for quota file attributes.
;   101          0101  1 !
;   102          0102  1 !       V02-002 MLJ0035         Martin L. Jack, 27-Aug-1981  21:29
;   103          0103  1 !                 Allow extension file ID to point to CONTIN.SYS.
;   104          0104  1 !
;   105          0105  1 !       V02-001 MLJ0030         Martin L. Jack, 30-Jul-1981  23:23
;   106          0106  1 !                 Consider alternate index file header invalid only if it fails
;   107          0107  1 !                 basic validation or if the map areas or EFBLK differ.  Change
;   108          0108  1 !                 listing default name string so that /LIST=LP: has a reasonable
;   109          0109  1 !                 file name.  Write-access the quota file to flush the ACP cache.
;   110          0110  1 !
;   111          0111  1 !**
;   112          0112  1
;   113          0113  1
;   114          0114  1 LIBRARY 'SYS$LIBRARY:LIB';
```

```
;  116     0115  1 LITERAL
;  117     0116  1         TRUE=           1,
;  118     0117  1         FALSE=          0;
;  119     0118  1
;  120     0119  1
;  121     0120  1 STRUCTURE
;  122     0121  1         BBLOCK[O,P,S,E;N]=
;  123     0122  1             [N]
;  124     0123  1             (BBLOCK + O)<P,S,E>;
;  125     0124  1
;  126     0125  1
;  127     0126  1 MACRO
;  128   M 0127  1         FAO_(A)=
;  129   M 0128  1             FAO(
;  130   M 0129  1                 UPLIT BYTE (%ASCIC A)
;  131     0130  1                 %IF NOT %NULL(%REMAINING) %THEN , %FI %REMAINING) %;
;  132     0131  1
;  133     0132  1
;  134     0133  1 LITERAL
;  135     0134  1         LIST_SIZE=      132;            ! Size of listing buffer
;  136     0135  1
;  137     0136  1
;  138     0137  1 SWITCHES
;  139     0138  1         ADDRESSING_MODE(
;  140     0139  1             EXTERNAL=LONG_RELATIVE,
;  141     0140  1             NONEXTERNAL=WORD_RELATIVE);
;  142     0141  1
;  143     0142  1
;  144     0143  1 PSECT
;  145     0144  1         CODE=           CODE,
;  146     0145  1         PLIT=           CODE,
;  147     0146  1         OWN=            DATA(ADDRESSING_MODE(LONG_RELATIVE)),
;  148     0147  1         GLOBAL=         DATA;
;  149     0148  1
;  150     0149  1
;  151     0150  1 MACRO
;  152     0151  1
;  153     0152  1         ! Field definitions for QUAL area, general qualifiers.
;  154     0153  1         !
;  155     0154  1         QUAL_CONF=      0,0,1,0 %,      ! /CONFIRM
;  156     0155  1         QUAL_LIST=      0,1,1,0 %,      ! /LIST
;  157     0156  1         QUAL_READ=      0,2,1,0 %,      ! /READ_CHECK
;  158     0157  1         QUAL_REPA=      0,3,1,0 %,      ! /REPAIR
;  159     0158  1         QUAL_USAG=      0,4,1,0 %;      ! /USAGE
```

```
;  161     0159  1 FORWARD ROUTINE
;  162     0160  1           VERIFY,                          ! Main routine
;  163     0161  1           INIT_VOL_DATA:  NOVALUE,         ! Initialize per-volume data
;  164     0162  1           READ_HOMEBLOCK: NOVALUE,         ! Read and check home block
;  165     0163  1           SCAN_INDEX:     NOVALUE,         ! Scan index files
;  166     0164  1           VERIFY_HEADER,                   ! Check a file header
;  167     0165  1           MAP_PROCESS:    NOVALUE,         ! Process map area
;  168     0166  1           CREATE_WINDOW,                   ! Create window for file
;  169     0167  1           DELETE_WINDOW:  NOVALUE,         ! Delete window for file
;  170     0168  1           MAP_VIRTUAL,                     ! Map virtual to logical for file
;  171     0169  1           ACCESS_INDEX_2: NOVALUE,         ! Access index file on second channel
;  172     0170  1           COUNT_QUOTA:    NOVALUE,         ! Maintain quota data base
;  173     0171  1           READ_HEADER,                     ! Read a file header
;  174     0172  1           WRITE_HEADER,                    ! Write a file header
;  175     0173  1           DELETE_HEADER:  NOVALUE,         ! Write a deleted file header
;  176     0174  1           CLEAR_EXT_FID:  NOVALUE,         ! Clear extension linkage
;  177     0175  1           READ_CHECK:     NOVALUE,         ! Do read checking
;  178     0176  1           FILE_ERROR:     NOVALUE,         ! Signal a file-related error
;  179     0177  1           HEADER_ERROR:   NOVALUE,         ! Signal a header-related error
;  180     0178  1           PROCESS_FILE,                    ! Process one file
;  181     0179  1           PROCESS_SUBDIR: NOVALUE,         ! Process one subdirectory
;  182     0180  1           SCAN_DIRECT_1,                   ! Scan a directory (ODS-1)
;  183     0181  1           SCAN_DIRECT_2,                   ! Scan a directory (ODS-2)
;  184     0182  1           DIR_SCAN:       NOVALUE,         ! Scan all directories
;  185     0183  1           FAO:            NOVALUE,         ! Format information to listing
;  186     0184  1           EOL:            NOVALUE,         ! Write listing buffer to file
;  187     0185  1           ENTER_WORK:     NOVALUE,         ! Enter an item on work list
;  188     0186  1           PROCESS_WORK:   NOVALUE,         ! Process work list
;  189     0187  1           DO_REPAIR,                       ! Evaluate repair control qualifiers
;  190     0188  1           EXIT_HANDLER:   NOVALUE,         ! Exit handler
;  191     0189  1           CTRL_AST:       NOVALUE,         ! CTRL/C, CTRL/Y handler
;  192     0190  1           CHECK_DATE:     NOVALUE;         ! Check date field
;  193     0191  1
;  194     0192  1
;  195     0193  1 EXTERNAL ROUTINE
;  196     0194  1           CHECKSUM,                        ! Compute file header checksum
;  197     0195  1           CHECKSUM2,                       ! Compute home block checksum
;  198     0196  1           LEFT_ONE,                        ! Find leftmost one bit in a longword
;  199     0197  1           MAKE_STRING,                     ! Convert ODS-1 filename to ASCII
;  200     0198  1           CLI$GET_VALUE:  ADDRESSING_MODE(GENERAL),
;  201     0199  1                                            ! Get a parameter or qualifier value
;  202     0200  1           CLI$PRESENT:    ADDRESSING_MODE(GENERAL),
;  203     0201  1                                            ! Determine if entity is present
;  204     0202  1           LIB$DISABLE_CTRL: ADDRESSING_MODE(GENERAL),
;  205     0203  1                                            ! Disable CTRL/x handling
;  206     0204  1           LIB$ENABLE_CTRL: ADDRESSING_MODE(GENERAL),
;  207     0205  1                                            ! Re-enable it again
;  208     0206  1           LIB$FREE_VM:    ADDRESSING_MODE(GENERAL),
;  209     0207  1                                            ! Free virtual memory
;  210     0208  1           LIB$GET_COMMAND:ADDRESSING_MODE(GENERAL),
;  211     0209  1                                            ! Get line from SYS$COMMAND
;  212     0210  1           LIB$GET_VM:     ADDRESSING_MODE(GENERAL),
;  213     0211  1                                            ! Get virtual memory
;  214     0212  1           LIB$SCOPY_R_DX: ADDRESSING_MODE(GENERAL),
;  215     0213  1                                            ! Copy a string
;  216     0214  1           LIB$SIGNAL:     ADDRESSING_MODE(GENERAL);
;  217     0215  1                                            ! Signal a condition
```

```
;  218     0216  1
;  219     0217  1
;  220     0218  1 EXTERNAL LITERAL
;  221     0219  1         VERIFY$_FACILITY,
;  222     0220  1         VERIFY$_ABORT,
;  223     0221  1         VERIFY$_ADDQUOTA,
;  224     0222  1         VERIFY$_ALLOCCLR,
;  225     0223  1         VERIFY$_ALLOCEXT,
;  226     0224  1         VERIFY$_ALLOCMEM,
;  227     0225  1         VERIFY$_ALLOCSET,
;  228     0226  1         VERIFY$_ALTIHDBAD,
;  229     0227  1         VERIFY$_ASSIGN,
;  230     0228  1         VERIFY$_BACKLINK,
;  231     0229  1         VERIFY$_BADBITMAP,
;  232     0230  1         VERIFY$_BADDIR,
;  233     0231  1         VERIFY$_BADDIRENT,
;  234     0232  1         VERIFY$_BADEFBLK,
;  235     0233  1         VERIFY$_BADHEADER,
;  236     0234  1         VERIFY$_BADHIBLK,
;  237     0235  1         VERIFY$_BBLHEADER,
;  238     0236  1         VERIFY$_CHKALTHOME,
;  239     0237  1         VERIFY$_CHKPRIHOME,
;  240     0238  1         VERIFY$_CHKSCB,
;  241     0239  1         VERIFY$_CREATELOST,
;  242     0240  1         VERIFY$_DELETE,
;  243     0241  1         VERIFY$_DELHEADER,
;  244     0242  1         VERIFY$_DIRNAME,
;  245     0243  1         VERIFY$_DSAQUOTA,
;  246     0244  1         VERIFY$_ENAQUOTA,
;  247     0245  1         VERIFY$_ENTERLOST,
;  248     0246  1         VERIFY$_FINDHOME,
;  249     0247  1         VERIFY$_FINDIHD,
;  250     0248  1         VERIFY$_FREEMEM,
;  251     0249  1         VERIFY$_FUTBAKDAT,
;  252     0250  1         VERIFY$_FUTCREDAT,
;  253     0251  1         VERIFY$_FUTREVDAT,
;  254     0252  1         VERIFY$_GETDVI,
;  255     0253  1         VERIFY$_INCQUOTA,
;  256     0254  1         VERIFY$_INVDEVICE,
;  257     0255  1         VERIFY$_INVEXTBACK,
;  258     0256  1         VERIFY$_INVEXTFID,
;  259     0257  1         VERIFY$_INVEXTHDR,
;  260     0258  1         VERIFY$_LOCKHEADER,
;  261     0259  1         VERIFY$_LOCKVOL,
;  262     0260  1         VERIFY$_LOSTEXTHDR,
;  263     0261  1         VERIFY$_LOSTHEADER,
;  264     0262  1         VERIFY$_LOSTSCAN,
;  265     0263  1         VERIFY$_MAPAREA,
;  266     0264  1         VERIFY$_MAXVOLS,
;  267     0265  1         VERIFY$_MODQUOTA,
;  268     0266  1         VERIFY$_MULTALLOC,
;  269     0267  1         VERIFY$_MULTEXTHDR,
;  270     0268  1         VERIFY$_NOREPAIR,
;  271     0269  1         VERIFY$_OPENBITMAP,
;  272     0270  1         VERIFY$_OPENDIR,
;  273     0271  1         VERIFY$_OPENFILE,
;  274     0272  1         VERIFY$_OPENINDEX,
```

```
;  275     0273  1          VERIFY$_OPENQUOTA,
;  276     0274  1          VERIFY$_PRIIHDBAD,
;  277     0275  1          VERIFY$_READBOOT,
;  278     0276  1          VERIFY$_READDIR,
;  279     0277  1          VERIFY$_READFILE,
;  280     0278  1          VERIFY$_READHEADER,
;  281     0279  1          VERIFY$_READHOME,
;  282     0280  1          VERIFY$_READIBMAP,
;  283     0281  1          VERIFY$_READQUOTA,
;  284     0282  1          VERIFY$_READSBMAP,
;  285     0283  1          VERIFY$_READSCB,
;  286     0284  1          VERIFY$_REMOVE,
;  287     0285  1          VERIFY$_UNLKVOL,
;  288     0286  1          VERIFY$_WRITEHEADER,
;  289     0287  1          VERIFY$_WRITEHOME,
;  290     0288  1          VERIFY$_WRITEIBMAP,
;  291     0289  1          VERIFY$_WRITESBMAP,
;  292     0290  1          VERIFY$_WRITESCB,
;  293     0291  1          VERIFY$_WRONGOWNER;
;  294     0292  1
;  295     0293  1
;  296     0294  1 LITERAL
;  297     0295  1          MAX_VOLUMES=    255,          ! Largest volume set
;  298     0296  1          DIR_BUF_COUNT=  16,           ! Size of directory buffer
;  299     0297  1          INDEX_BUF_COUNT=64,           ! Size of index file buffer
;  300     0298  1          FILE_BUF_COUNT= 64;           ! Size of file data buffer
;  301     0299  1
;  302     0300  1
;  303     0301  1 MACRO
;  304     0302  1          DVI_MAXBLOCK=   0,0,32,0 %,   ! DVI$_MAXBLOCK
;  305     0303  1          DVI_SECTORS=    4,0,32,0 %,   ! DVI$_SECTORS
;  306     0304  1          DVI_TRACKS=     8,0,32,0 %,   ! DVI$_TRACKS
;  307     0305  1          DVI_CYLINDERS=  12,0,32,0 %;  ! DVI$_CYLINDERS
;  308     0306  1
;  309     0307  1 LITERAL
;  310     0308  1          DVI_LENGTH=     16;           ! Length of DEVICE_CHAR area
;  311     0309  1
;  312     0310  1
;  313     0311  1 OWN
;  314     0312  1          QUAL:           BBLOCK[4],                          ! Qualifier bits
;  315     0313  1          QUAL_DEV_DESC:  BBLOCK[8],                          ! Value of device parameter
;  316     0314  1          QUAL_LIST_DESC: BBLOCK[8],                          ! Value of /LIST qualifier
;  317     0315  1          QUAL_USAG_DESC: BBLOCK[8],                          ! Value of /USAGE qualifier
;  318     0316  1          LIST_FAB:       BBLOCK[FAB$C_BLN],                  ! FAB for listing file
;  319     0317  1          LIST_RAB:       BBLOCK[RAB$C_BLN],                  ! RAB for listing file
;  320     0318  1          LIST_NAM:       BBLOCK[NAM$C_BLN],                  ! NAM block for listing file
;  321     0319  1          LIST_RSA:       VECTOR[NAM$C_MAXRSS,BYTE],          ! Resultant string for listing file
;  322     0320  1          LIST_DESC:      VECTOR[2],                          ! Descriptor for listing buffer
;  323     0321  1          LIST_BUFFER:    VECTOR[LIST_SIZE,BYTE],             ! Listing line buffer
;  324     0322  1          USAGE_FAB:      BBLOCK[FAB$C_BLN],                  ! FAB for usage file
;  325     0323  1          USAGE_RAB:      BBLOCK[RAB$C_BLN],                  ! RAB for usage file
;  326     0324  1          USAGE_NAM:      BBLOCK[NAM$C_BLN],                  ! NAM block for usage file
;  327     0325  1          USAGE_RSA:      VECTOR[NAM$C_MAXRSS,BYTE],          ! Resultant string for usage file
;  328     0326  1          USAGE_BUFFER:   BBLOCK[MAXU(USG$C_IDENT_LEN,USG$C_FILE_LEN)],
;  329     0327  1                                                              ! Buffer for usage file
;  330     0328  1          CHANNEL_TT,                                         ! Channel for SYS$COMMAND
;  331     0329  1          CHANNEL,                                            ! Channel to device
```

```
;  332     0330  1        CHANNEL_2,                                  ! Second channel to device
;  333     0331  1        CHAN2_RVN,                                  ! If nonzero, index file on this RVN is accessed on
;  334     0332  1        TOTAL_SIZE,                                 ! Total space mapped by headers for current file
;  335     0333  1        DUAL_ALLOC_FOUND,                           ! True if multiply allocated blocks exist
;  336     0334  1        DUAL_ALLOC_PASS,                            ! True if second pass to find multiply allocated blo
;  337     0335  1        DIRECTORY_ERROR,                            ! True if error occurred during directory scan
;  338     0336  1        OLD_CTRL_MASK,                              ! Old ctrl char enable mask
;  339     0337  1        EXIT_HAND_DESC: VECTOR[5],                  ! Exit handler descriptor
;  340     0338  1        QT:             BBLOCK[8],                  ! Head of quota table list
;  341     0339  1        QUOTA_ACTIVE,                               ! True if quota file on volume set
;  342     0340  1        QUOTA_DISABLE,                              ! True if quota processing must be disabled
;  343     0341  1        DEFAULT_QUOTA,                              ! Default quota value
;  344     0342  1        DEFAULT_OVERDRAFT,                          ! Default overdraft value
;  345     0343  1        LAST_UIC,                                   ! Last existing UIC in quota file
;  346     0344  1        DQF:            BBLOCK[DQF$C_LENGTH],       ! Quota record buffer
;  347     0345  1        BUFFER:         BBLOCK[512*INDEX_BUF_COUNT], ! Index file buffer
;  348     0346  1        BUFFER_2:       BBLOCK[512],                ! Second buffer
;  349     0347  1        FIB:            BBLOCK[FIB$C_LENGTH],       ! General FIB used for all QIO's
;  350     0348  1        RECATTR:        BBLOCK[FAT$C_LENGTH],       ! Record attributes area
;  351     0349  1        UCHAR:          BBLOCK[4],                  ! File characteristics area
;  352     0350  1        DIR:            VECTOR[320,BYTE],           ! Current directory string
;  353     0351  1        DIR_DESC:       VECTOR[2],                  ! Descriptor for current directory string
;  354     0352  1        DIR_FID:        BBLOCK[FID$C_LENGTH],       ! File ID of current directory
;  355     0353  1        LOST_DIR_FID:   BBLOCK[FID$C_LENGTH],       ! File ID of lost file directory
;  356     0354  1        IOSB:           VECTOR[4,WORD],             ! General I/O status block used for all QIO's
;  357     0355  1        VOLUME_COUNT,                               ! Number of volumes in volume set
;  358     0356  1        STRUCTURE_LEVEL,                            ! Structure level of volume set
;  359     0357  1        HOMEVBN,                                    ! VBN of primary home block
;  360     0358  1        WORK_LIST:      VECTOR[2],                  ! Work list header
;  361     0359  1        CURRENT_TIME:   VECTOR[2],                  ! Current time in 64-bit format
;  362     0360  1        CURRENT_TIME_1: BBLOCK[13],                 ! Current time in ODS-1 format
;  363     0361  1        DEVICE_DESC:    VECTOR[2],                  ! Descriptor for DEVICE_NAME
;  364     0362  1        DEVICE_NAME:    VECTOR[16,BYTE],            ! Device name for specific RVN
;  365     0363  1        DEVICE_CHAR:    BBLOCK[DVI_LENGTH],         ! Device characteristics buffer
;  366     0364  1
;  367     0365  1
;  368     0366  1        PER_VOLUME_BEG: VECTOR[0],                  ! Beginning of per-volume information
;  369     0367  1        ACCTL:          REF VECTOR,                 ! FIB$M_WRITE if write-accessing files
;  370     0368  1        IMAP_SIZE:      REF VECTOR,                 ! Number of blocks in index file bitmap
;  371     0369  1        MAXFILIDX:      REF VECTOR,                 ! Highest valid file number minus one
;  372     0370  1        IMAP:           REF VECTOR,                 ! Bitmap of valid file numbers = new index file bitm
;  373     0371  1        DIRMAP:         REF VECTOR,                 ! Bitmap of directory files
;  374     0372  1        SEQMAP:         REF VECTOR,                 ! Word vector of file sequence number
;  375     0373  1        BACKMAP:        REF VECTOR,                 ! Three-word vector of file back link FID
;  376     0374  1        LOSTMAP:        REF VECTOR,                 ! Bitmap of valid file numbers not yet found in dire
;  377     0375  1        EXTMAP:         REF VECTOR,                 ! Bitmap of file numbers referenced by extension lin
;  378     0376  1        OWNER:          REF VECTOR,                 ! Longword vector of file owner UIC
;  379     0377  1        ALLOCATION:     REF VECTOR,                 ! Longword vector of file allocated blocks
;  380     0378  1        USAGE:          REF VECTOR,                 ! Longword vector of file used blocks
;  381     0379  1
;  382     0380  1        SMAP_SIZE:      REF VECTOR,                 ! Number of blocks in storage bitmap
;  383     0381  1        VSMAP:          REF VECTOR,                 ! Bitmap of allocated clusters
;  384     0382  1        NSMAP:          REF VECTOR,                 ! VSMAP less lost extension headers = new storage bi
;  385     0383  1        MULTSMAP:       REF VECTOR,                 ! Bitmap of multiply allocated clusters
;  386     0384  1
;  387     0385  1        CLUSTER_FACTOR: REF VECTOR,                 ! Cluster factor
;  388     0386  1        HEADER_OFFSET:  REF VECTOR,                 ! VBN offset to file header
```

```
;  389     0387  1          BITMAP_OFFSET:  REF VECTOR,                  ! VBN offset to index file bitmap
;  390     0388  1          EOF:           REF VECTOR,                  ! VBN of index file EOF
;  391     0389  1          PER_VOLUME_END: VECTOR[0];                  ! End of per-volume information
;  392     0390  1
;  393     0391  1
;  394     0392  1  LITERAL
;  395     0393  1
;  396     0394  1          ! Values of the parameter to DO_REPAIR.
;  397     0395  1          !
;  398     0396  1          NO_CONFIRM=    %B'00',      ! /CONFIRM prompting inhibited
;  399     0397  1          ALLOW_DELETE=  %B'11';      ! DELETE is a /CONFIRM option
;  400     0398  1
;  401     0399  1
;  402     0400  1  MACRO
;  403     0401  1
;  404     0402  1          ! Field definitions for work list.
;  405     0403  1          !
;  406     0404  1          WRK_LINK=      0,0,32,0 %,  ! Link to next block
;  407     0405  1          WRK_TYPE=      4,0,8,0 %,   ! Type of entry
;  408     0406  1          WRK_FID=       6,0,16,0 %,  ! File ID of entry      (ENT, REM, DEL)
;  409     0407  1          WRK_DID=       12,0,16,0 %, ! Directory ID of entry (REM)
;  410     0408  1          WRK_UIC=       8,0,32,0 %,  ! UIC of entry          (ADD)
;  411     0409  1          WRK_USAGE=     12,0,32,0 %; ! Usage of entry        (ADD)
;  412     0410  1
;  413     0411  1
;  414     0412  1  LITERAL
;  415     0413  1          WRK_S_ENTER=   12,          ! Size of ENTER entry
;  416     0414  1          WRK_S_REMOVE=  18,          ! Size of REMOVE entry
;  417     0415  1          WRK_S_ADDQUO=  16,          ! Size of ADDQUO entry
;  418     0416  1          WRK_S_DELETE=  12,          ! Size of DELETE entry
;  419     0417  1          WRK_K_ENTER=   0,           ! Enter file in [SYSLOST]
;  420     0418  1          WRK_K_REMOVE=  1,           ! Remove file from directory
;  421     0419  1          WRK_K_ADDQUO=  2,           ! Add quota entry
;  422     0420  1          WRK_K_DELETE=  3;           ! Delete file
;  423     0421  1
;  424     0422  1
;  425     0423  1  MACRO
;  426     0424  1
;  427     0425  1          ! Field definitions for quota table.
;  428     0426  1          !
;  429     0427  1          QT_LINK=       0,0,32,0 %,  ! Link to next block
;  430     0428  1          QT_COUNT=      4,0,32,0 %,  ! Count of used entries in this block
;  431     0429  1
;  432     0430  1          QT_UIC=        0,0,32,0 %,  ! Entry UIC
;  433     0431  1          QT_QUO_USED=   4,0,32,0 %,  ! Entry usage per quota file
;  434     0432  1          QT_IDX_USED=   8,0,32,0 %;  ! Entry usage per index file
;  435     0433  1
;  436     0434  1
;  437     0435  1  LITERAL
;  438     0436  1          QT_S_HDR=      8,           ! Size of quota table block header
;  439     0437  1          QT_S_ENT=      12,          ! Size of quota table entry
;  440     0438  1          QT_MAXCOUNT=   256;         ! Maximum entries per block
;  441     0439  1
;  442     0440  1
;  443     0441  1  MACRO
;  444     0442  1
;  445     0443  1          ! Field definitions for window block.
```

```
;  446     0444  1          !
;  447     0445  1          WDW_LINK=       0,0,32,0 %,     ! Link to next block
;  448     0446  1          WDW_SIZE=       4,0,32,0 %,     ! Number of entries
;  449     0447  1          WDW_ENTRY=      8,0,0,0 %,      ! Beginning of first entry
;  450     0448  1
;  451     0449  1          WDW_COUNT=      0,0,32,0 %,     ! Count of blocks
;  452     0450  1          WDW_LBN=        4,0,32,0 %;     ! LBN of blocks
;  453     0451  1
;  454     0452  1
;  455     0453  1  LITERAL
;  456     0454  1          WDW_S_HEADER=   8,              ! Size of window block header
;  457     0455  1          WDW_S_ENTRY=    8,              ! Size of window block entry
;  458     0456  1          WDW_K_MAXENTRY= 16;             ! Maximum number of entries in each block
;  459     0457  1
;  460     0458  1
;  461     0459  1  OWN
;  462     0460  1          LOST_NAME:      VECTOR[13,BYTE] INITIAL (BYTE('SYSLOST.DIR;1')),
;  463     0461  1          QFI_NAME:       VECTOR[11,BYTE] INITIAL (BYTE('QUOTA.SYS;1'));
;  464     0462  1
;  465     0463  1
;  466     0464  1  BIND
;  467     0465  1          HDR_BUFFER=     BUFFER + 512: BBLOCK,
;  468     0466  1          HDR_BUFFER_2=   BUFFER + 1024: BBLOCK,
;  469     0467  1          MFD_DESC=       $DESCRIPTOR('000000'),
;  470     0468  1          FAT_ATR_DESC=   UPLIT(
;  471     0469  1                              WORD(ATR$S_RECATTR, ATR$C_RECATTR), RECATTR,
;  472     0470  1                              WORD(ATR$S_UCHAR, ATR$C_UCHAR), UCHAR,
;  473     0471  1                              0),
;  474     0472  1          HDR_ATR_DESC=   UPLIT(
;  475     0473  1                              WORD(ATR$S_HEADER, ATR$C_HEADER), HDR_BUFFER,
;  476     0474  1                              0),
;  477     0475  1          FIB_DESC=       UPLIT(FIB$C_LENGTH, FIB),
;  478     0476  1          DQF_DESC=       UPLIT(DQF$C_LENGTH, DQF),
;  479     0477  1          LOST_DESC=      UPLIT(13, LOST_NAME),
;  480     0478  1          QFI_DESC=       UPLIT(11, QFI_NAME);
;  481     0479  1
;  482     0480  1
;  483     0481  1  BUILTIN
;  484     0482  1          CALLG,
;  485     0483  1          EDIV,
;  486     0484  1          ROT,
;  487     0485  1          TESTBITSS,
;  488     0486  1          TESTBITSC,
;  489     0487  1          TESTBITCS,
;  490     0488  1          TESTBITCC;
```

```
;  492          0489  1 ROUTINE VERIFY=
;  493          0490  1
;  494          0491  1 !++
;  495          0492  1 !
;  496          0493  1 ! FUNCTIONAL DESCRIPTION:
;  497          0494  1 !       This routine is the main entry point to the VERIFY utility.
;  498          0495  1 !
;  499          0496  1 ! INPUT PARAMETERS:
;  500          0497  1 !       Standard VMS activation parameters (not used).
;  50.          0498  1 !
;  502          0499  1 ! IMPLICIT INPUTS:
;  503          0500  1 !       NONE
;  504          0501  1 !
;  505          0502  1 ! OUTPUT PARAMETERS:
;  506          0503  1 !       NONE
;  507          0504  1 !
;  508          0505  1 ! IMPLICIT OUTPUTS:
;  509          0506  1 !       NONE
;  510          0507  1 !
;  511          0508  1 ! ROUTINE VALUE:
;  512          0509  1 !       SS$_NORMAL.
;  513          0510  1 !
;  514          0511  1 ! SIDE EFFECTS:
;  515          0512  1 !       NONE
;  516          0513  1 !
;  517          0514  1 !--
;  518          0515  1
;  519          0516  2 BEGIN
;  520          0517  2 LOCAL
;  521          0518  2         ACCTL_0,                        ! ACCTL[0] before ACCTL is allocated
;  522          0519  2         NO_WRITE,                       ! True if any volume is write locked
;  523          0520  2         STATUS;                         ! General status variable
;  524          0521  2
;  525          0522  2
;  526          0523  2 ! Get the parameter, the device to be verified.
;  527          0524  2 !
;  528          0525  2 QUAL_DEV_DESC[DSC$B_CLASS] = DSC$K_CLASS_D;
;  529          0526  2 CLI$GET_VALUE($DESCRIPTOR('DEVICE'), QUAL_DEV_DESC);
;  530          0527  2
;  531          0528  2
;  532          0529  2 ! Execute a $PARSE to check the specification.  It must have a valid device,
;  533          0530  2 ! but must not have any other file specification components.  The listing FAB
;  534          0531  2 ! is used as a temporary for this operation.
;  535          0532  2 !
;  536        P 0533  2 $FAB_INIT(FAB=LIST_FAB,
;  537        P 0534  2       FNA=.QUAL_DEV_DESC[DSC$A_POINTER],
;  538        P 0535  2       FNS=.QUAL_DEV_DESC[DSC$W_LENGTH],
;  539          0536  2       NAM=LIST_NAM);
;  540        P 0537  2 $NAM_INIT(NAM=LIST_NAM,
;  541        P 0538  2       ESA=LIST_RSA,
;  542          0539  2       ESS=NAM$C_MAXRSS);
;  543          0540  2 $PARSE(FAB=LIST_FAB);
;  544          0541  2 IF
;  545          0542  2       .LIST_NAM[NAM$B_ESL] EQL 0 OR
;  546          0543  3       (.LIST_NAM[NAM$C_FNB] AND
;  547          0544  4           (NAM$M_EXP_DIR OR
;  548          0545  4            NAM$M_EXP_NAME OR
```

```
;  549          0546  4            NAM$M_EXP_TYPE OR
;  550          0547  4            NAM$M_EXP_VER OR
;  551          0548  4            NAM$M_NODE OR
;  552          0549  2            NAM$M_QUOTED)) NEQ 0
;  553          0550  2 THEN
;  554          0551  2     SIGNAL(VERIFY$_INVDEVICE, 1, QUAL_DEV_DESC);
;  555          0552  2
;  556          0553  2
;  557          0554  2 ! Store the device name back into the device descriptor.
;  558          0555  2 !
;  559          0556  2 LIB$SCOPY_R_DX(
;  560          0557  2     %REF(.LIST_NAM[NAM$B_DEV]),
;  561          0558  2     LIST_RSA,
;  562          0559  2     QUAL_DEV_DESC);
;  563          0560  2
;  564          0561  2
;  565          0562  2 ! Initialize the device name descriptor with the device name of RVN 1.
;  566          0563  2 ! Ensure that the device is valid.
;  567          0564  2 !
;  568          0565  2 DEVICE_DESC[0] = 0;
;  569          0566  2 DEVICE_DESC[1] = DEVICE_NAME;
;  570        P 0567  2 STATUS = $GETDVI(
;  571        P 0568  2     DEVNAM=QUAL_DEV_DESC,
;  572        P 0569  2     ITMLST=UPLIT(
;  573        P 0570  2         WORD(4, DVI$_DEVCHAR OR DVI$C_SECONDARY),
;  574        P 0571  2         LONG(DEVICE_CHAR, 0),
;  575        P 0572  2         WORD(16, DVI$_ROOTDEVNAM),
;  576        P 0573  2         LONG(DEVICE_NAME, DEVICE_DESC),
;  577          0574  2         LONG(0)));
;  578          0575  2 IF NOT .STATUS
;  579          0576  2 THEN
;  580          0577  2     SIGNAL(VERIFY$_GETDVI, 1, 1, .STATUS);
;  581          0578  2
;  582          0579  2
;  583          0580  2 IF NOT .DEVICE_CHAR[DEV$V_RND]
;  584          0581  2 THEN
;  585          0582  2     SIGNAL(VERIFY$_INVDEVICE, 1, QUAL_DEV_DESC);
;  586          0583  2
;  587          0584  2
;  588          0585  2 ! Get value of /LIST.
;  589          0586  2 !
;  590          0587  2 QUAL_LIST_DESC[DSC$B_CLASS] = DSC$K_CLASS_D;
;  591          0588  2 IF CLI$PRESENT($DESCRIPTOR('LIST'))
;  592          0589  2 THEN
;  593          0590  3     BEGIN
;  594          0591  3     QUAL[QUAL_LIST] = TRUE;
;  595          0592  3     CLI$GET_VALUE($DESCRIPTOR('LIST'), QUAL_LIST_DESC);
;  596          0593  2     END;
;  597          0594  2
;  598          0595  2
;  599          0596  2 ! Get value of /USAGE.
;  600          0597  2 !
;  601          0598  2 QUAL_USAG_DESC[DSC$B_CLASS] = DSC$K_CLASS_D;
;  602          0599  2 IF CLI$PRESENT($DESCRIPTOR('USAGE'))
;  603          0600  2 THEN
;  604          0601  3     BEGIN
;  605          0602  3     QUAL[QUAL_USAG] = TRUE;
```

```
;  606     0603  3     CLI$GET_VALUE($DESCRIPTOR('USAGE'), QUAL_USAG_DESC);
;  607     0604  3     END;
;  608     0605  3
;  609     0606  3
;  610     0607  3     ! Get value of Boolean qualifiers.
;  611     0608  3     !
;  612     0609  3     QUAL[QUAL_CONF] = CLI$PRESENT($DESCRIPTOR('CONFIRM'));
;  613     0610  3     QUAL[QUAL_READ] = CLI$PRESENT($DESCRIPTOR('READ_CHECK'));
;  614     0611  3     QUAL[QUAL_REPA] = CLI$PRESENT($DESCRIPTOR('REPAIR'));
;  615     0612  3
;  616     0613  3
;  617     0614  3     ! Open the listing file.
;  618     0615  3     !
;  619     0616  3     IF .QUAL[QUAL_LIST]
;  620     0617  3     THEN
;  621     0618  3         BEGIN
;  622   P 0619  3         $FAB_INIT(FAB=LIST_FAB,
;  623   P 0620  3             DNA=UPLIT BYTE('VERIFY.LIS'),
;  624   P 0621  3             DNS=%CHARCOUNT('VERIFY.LIS'),
;  625   P 0622  3             FAC=PUT,
;  626   P 0623  3             FNA=.QUAL_LIST_DESC[DSC$A_POINTER],
;  627   P 0624  3             FNS=.QUAL_LIST_DESC[DSC$W_LENGTH],
;  628   P 0625  3             FOP=SQO,
;  629   P 0626  3             NAM=LIST_NAM,
;  630   P 0627  3             ORG=SEQ,
;  631   P 0628  3             RAT=CR,
;  632     0629  3             RFM=VAR);
;  633   P 0630  3         $RAB_INIT(RAB=LIST_RAB,
;  634   P 0631  3             FAB=LIST_FAB,
;  635   P 0632  3             RBF=LIST_BUFFER,
;  636     0633  3             ROP=WBH);
;  637   P 0634  3         $NAM_INIT(NAM=LIST_NAM,
;  638   P 0635  3             ESA=LIST_RSA,
;  639   P 0636  3             ESS=NAM$C_MAXRSS,
;  640   P 0637  3             RSA=LIST_RSA,
;  641     0638  3             RSS=NAM$C_MAXRSS);
;  642     0639  3         IF .LIST_FAB[FAB$B_FNS] EQL 0
;  643     0640  3         THEN
;  644     0641  4             BEGIN
;  645     0642  4             LIST_FAB[FAB$B_FNS] = %CHARCOUNT('SYS$OUTPUT:');
;  646     0643  4             LIST_FAB[FAB$L_FNA] = UPLIT BYTE('SYS$OUTPUT:');
;  647     0644  4             END;
;  648     0645  3         LIST_DESC[0] = LIST_SIZE;
;  649     0646  3         LIST_DESC[1] = LIST_BUFFER;
;  650     0647  3         IF NOT $CREATE(FAB=LIST_FAB)
;  651     0648  3         THEN
;  652     0649  3             FILE_ERROR(
;  653     0650  3                 VERIFY$_FACILITY^16 + SHR$_OPENOUT + STS$K_SEVERE,
;  654     0651  3                 LIST_FAB,
;  655     0652  3                 .LIST_FAB[FAB$L_STS], .LIST_FAB[FAB$L_STV]);
;  656     0653  4         IF NOT $CONNECT(RAB=LIST_RAB)
;  657     0654  4         THEN
;  658     0655  4             FILE_ERROR(
;  659     0656  4                 VERIFY$_FACILITY^16 + SHR$_OPENOUT + STS$K_SEVERE,
;  660     0657  4                 LIST_FAB,
;  661     0658  4                 .LIST_RAB[RAB$L_STS], .LIST_RAB[RAB$L_STV]);
;  662     0659  3
```

```
;  663     0660  !
;  664     0661  !       Put a header line on the listing.
;  665     0662  !
;  666     0663  !       FAO ('Listing of index file on !AS!/!%D', QUAL_DEV_DESC, 0);
;  667     0664  !       EOL();
;  668     0665  !       EOL();
;  669     0666  !       END;
;  670     0667  !
;  671     0668  !
;  672     0669  ! Open the usage file.
;  673     0670  !
;  674     0671  IF .QUAL[QUAL_USAG]
;  675     0672  THEN
;  676     0673      BEGIN
;  677   P 0674      $FAB_INIT(FAB=USAGE_FAB,
;  678   P 0675          DNA=UPLIT BYTE('USAGE.DAT'),
;  679   P 0676          DNS=%CHARCOUNT('USAGE.DAT'),
;  680   P 0677          FAC=PUT,
;  681   P 0678          FNA=.QUAL_USAG_DESC[DSC$A_POINTER],
;  682   P 0679          FNS=.QUAL_USAG_DESC[DSC$W_LENGTH],
;  683   P 0680          FOP=SQO,
;  684   P 0681          NAM=USAGE_NAM,
;  685   P 0682          ORG=SEQ,
;  686   P 0683          RAT=CR,
;  687     0684          RFM=VAR);
;  688   P 0685      $RAB_INIT(RAB=USAGE_RAB,
;  689   P 0686          FAB=USAGE_FAB,
;  690   P 0687          RBF=USAGE_BUFFER,
;  691     0688          ROP=WBH);
;  692   P 0689      $NAM_INIT(NAM=USAGE_NAM,
;  693   P 0690          ESA=USAGE_RSA,
;  694   P 0691          ESS=NAM$C_MAXRSS,
;  695   P 0692          RSA=USAGE_RSA,
;  696     0693          RSS=NAM$C_MAXRSS);
;  697     0694      IF NOT $CREATE(FAB=USAGE_FAB)
;  698     0695      THEN
;  699     0696          FILE_ERROR(
;  700     0697              VERIFY$_FACILITY^16 + SHR$_OPENOUT + STS$K_SEVERE,
;  701     0698              USAGE_FAB,
;  702     0699              .USAGE_FAB[FAB$L_STS], .USAGE_FAB[FAB$L_STV]);
;  703     0700      IF NOT $CONNECT(RAB=USAGE_RAB)
;  704     0701      THEN
;  705     0702          FILE_ERROR(
;  706     0703              VERIFY$_FACILITY^16 + SHR$_OPENOUT + STS$K_SEVERE,
;  707     0704              USAGE_FAB,
;  708     0705              .USAGE_RAB[RAB$L_STS], .USAGE_RAB[RAB$L_STV]);
;  709     0706      END;
;  710     0707  !
;  711     0708  !
;  712     0709  ! Assign two channels to the device.
;  713     0710  !
;  714     0711  2 STATUS = $ASSIGN(DEVNAM=QUAL_DEV_DESC, CHAN=CHANNEL);
;  715     0712  2 IF NOT .STATUS THEN SIGNAL(VERIFY$_ASSIGN, 1, QUAL_DEV_DESC, .STATUS);
;  716     0713  2 STATUS = $ASSIGN(DEVNAM=QUAL_DEV_DESC, CHAN=CHANNEL_2);
;  717     0714  2 IF NOT .STATUS THEN SIGNAL(VERIFY$_ASSIGN, 1, QUAL_DEV_DESC, .STATUS);
;  718     0715  2 !
;  719     0716  2 ! Assign a channel to SYS$COMMAND, disable CLI ctrl/y handling, and
```

```
;  720     0717  2 ! enable ast's for CTRL/C and CTRL/Y.
;  721     0718  2 !
;  722   P 0719  2 STATUS = $ASSIGN(DEVNAM=$DESCRIPTOR('SYS$COMMAND'),
;  723     0720  2                   CHAN=CHANNEL_TT);
;  724     0721  2 IF .STATUS
;  725     0722  2 THEN BEGIN
;  726     0723  3     LIB$DISABLE_CTRL(%REF(%X'02000000'),OLD_CTRL_MASK);
;  727   P 0724  3     $QIOW(FUNC=(IO$_SETMODE OR IO$M_CTRLCAST),
;  728   P 0725  3         CHAN=.CHANNEL_TT,
;  729   P 0726  3         IOSB=IOSB,
;  730   P 0727  3         P1=CTRL_AST,
;  731   P 0728  3         P2=0,
;  732     0729  3         P3=PSL$C_USER);
;  733   P 0730  3     $QIOW(FUNC=(IO$_SETMODE OR IO$M_CTRLYAST),
;  734   P 0731  3         CHAN=.CHANNEL_TT,
;  735   P 0732  3         IOSB=IOSB,
;  736   P 0733  3         P1=CTRL_AST,
;  737   P 0734  3         P2=0,
;  738     0735  3         P3=PSL$C_USER);
;  739     0736  3     END;
;  740     0737  2
;  741     0738  2 ! Declare the exit handler.
;  742     0739  2 !
;  743     0740  2 EXIT_HAND_DESC[1] = EXIT_HANDLER;                   ! Handler address
;  744     0741  2 EXIT_HAND_DESC[2] = 1;                              ! Argument count
;  745     0742  2 EXIT_HAND_DESC[3] = EXIT_HAND_DESC[4];              ! Pointer to status longword
;  746     0743  2 $DCLEXH(DESBLK=EXIT_HAND_DESC);
;  747     0744  2
;  748     0745  2 IF .QUAL[QUAL_REPA]
;  749     0746  2 THEN
;  750     0747  3     BEGIN
;  751     0748  3     ! Lock the volume set.
;  752     0749  3     !
;  753     0750  3     CH$FILL(0, FIB$C_LENGTH, FIB);
;  754     0751  3     FIB[FIB$W_CNTRLFUNC] = FIB$C_LOCK_VOL;
;  755   P 0752  3     STATUS = $QIOW(
;  756   P 0753  3         FUNC=IO$_ACPCONTROL,
;  757   P 0754  3         CHAN=.CHANNEL,
;  758   P 0755  3         IOSB=IOSB,
;  759     0756  3         P1=FIB_DESC);
;  760     0757  3     IF .STATUS THEN STATUS = .IOSB[0];
;  761     0758  3     IF NOT .STATUS
;  762     0759  3     THEN
;  763     0760  4         BEGIN
;  764     0761  4         SIGNAL(VERIFY$_LOCKVOL, 0, .STATUS);
;  765     0762  4         QUAL[QUAL_REPA] = FALSE;
;  766     0763  4         END;
;  767     0764  3     END;
;  768     0765  2
;  769     0766  2
;  770     0767  2 ! Access the index file on RVN 1.  Read the file header into HDR_BUFFER.
;  771     0768  2 !
;  772     0769  2 CH$FILL(0, FIB$C_LENGTH, FIB);
;  773     0770  2 NO_WRITE = FALSE;
;  774     0771  2 FIB[FIB$L_ACCTL] = ACCTL_0 = FIB$M_WRITE OR FIB$M_NORECORD;
;  775     0772  2 FIB[FIB$W_FID_NUM] = FID$C_INDEXF;
;  776     0773  2 FIB[FIB$W_FID_SEQ] = FID$C_INDEXF;
```

```
;  777          0774  2 FIB[FIB$W_FID_RVN] = 1;
;  778        P 0775  2 STATUS = $QIOW(
;  779        P 0776  2     FUNC=IO$_ACCESS OR IO$M_ACCESS,
;  780        P 0777  2     CHAN=.CHANNEL,
;  781        P 0778  2     IOSB=IOSB,
;  782        P 0779  2     P1=FIB_DESC,
;  783          0780  2     P5=HDR_ATR_DESC);
;  784          0781  2 IF .STATUS THEN STATUS = .IOSB[0];
;  785          0782  2 IF .STATUS EQL SS$_WRITLCK
;  786          0783  2 THEN
;  787          0784  3     BEGIN
;  788          0785  3     CH$FILL(0, FIB$C_LENGTH, FIB);
;  789          0786  3     NO_WRITE = TRUE;
;  790          0787  3     FIB[FIB$L_ACCTL] = ACCTL_0 = FIB$M_NORECORD;
;  791          0788  3     FIB[FIB$W_FID_NUM] = FID$C_INDEXF;
;  792          0789  3     FIB[FIB$W_FID_SEQ] = FID$C_INDEXF;
;  793          0790  3     FIB[FIB$W_FID_RVN] = 1;
;  794        P 0791  3     STATUS = $QIOW(
;  795        P 0792  3         FUNC=IO$_ACCESS OR IO$M_ACCESS,
;  796        P 0793  3         CHAN=.CHANNEL,
;  797        P 0794  3         IOSB=IOSB,
;  798        P 0795  3         P1=FIB_DESC,
;  799          0796  3         P5=HDR_ATR_DESC);
;  800          0797  3     IF .STATUS THEN STATUS = .IOSB[0];
;  801          0798  3     END;
;  802          0799  2 IF NOT .STATUS THEN SIGNAL(VERIFY$_OPENINDEX, 1, 1, .STATUS);
;  803          0800  2
;  804          0801  2
;  805          0802  2 ! Read the home block on RVN 1.  Establish the size of the volume set and
;  806          0803  2 ! the structure level.
;  807          0804  2 !
;  808          0805  2 READ_HOMEBLOCK(1);
;  809          0806  2
;  810          0807  2
;  811          0808  2 ! Generate the usage file identification entry if required.
;  812          0809  2 !
;  813          0810  2 IF .QUAL[QUAL_USAG]
;  814          0811  2 THEN
;  815          0812  3     BEGIN
;  816          0813  3     USAGE_BUFFER[USG$B_TYPE] = USG$K_IDENT;
;  817          0814  3     CH$MOVE(
;  818          0815  3         $BYTEOFFSET(USG$Q_TIME) - $BYTEOFFSET(USG$L_SERIALNUM),
;  819          0816  3         BUFFER[HM2$L_SERIALNUM],
;  820          0817  3         USAGE_BUFFER[USG$L_SERIALNUM]);
;  821          0818  3     $GETTIM(TIMADR=USAGE_BUFFER[USG$Q_TIME]);
;  822          0819  3     USAGE_RAB[RAB$W_RSZ] = USG$K_IDENT_LEN;
;  823          0820  4     IF NOT $PUT(RAB=USAGE_RAB)
;  824          0821  3     THEN
;  825          0822  3         FILE_ERROR(
;  826          0823  3             VERIFY$_FACILITY + SHR$_WRITEERR + STS$K_SEVERE,
;  827          0824  3             USAGE_FAB,
;  828          0825  3             .USAGE_RAB[RAB$L_STS], .USAGE_RAB[RAB$L_STV]);
;  829          0826  3     END;
;  830          0827  2
;  831          0828  2
;  832          0829  2 ! Allocate the per-volume data.  Each variable between PER_VOLUME_BEG and
;  833          0830  2 ! PER_VOLUME_END is a pointer to a vector that varies with the number of
```

```
;  834          0831  2 ! volumes in the volume set.
;  835          0832  2 !
;  836          0833  2 INCRA A FROM PER_VOLUME_BEG TO PER_VOLUME_END-%UPVAL BY %UPVAL DO
;  837          0834  2     BEGIN
;  838          0835  2     STATUS = LIB$GET_VM(%REF(.VOLUME_COUNT*%UPVAL), .A);
;  839          0836  2     IF NOT .STATUS THEN SIGNAL(VERIFY$_ALLOCMEM, 0, .STATUS);
;  840          0837  2     CH$FILL(0, .VOLUME_COUNT*%UPVAL, ..A);
;  841          0838  2     END;
;  842          0839  2
;  843          0840  2
;  844          0841  2 ! Initialize the per-volume data for RVN 1.
;  845          0842  2 !
;  846          0843  2 INIT_VOL_DATA(1, .ACCTL_0);
;  847          0844  2 ACCTL[0] = .ACCTL_0;
;  848          0845  2
;  849          0846  2
;  850          0847  2 ! If this is a volume set, access the index file of each volume in the set
;  851          0848  2 ! and get the per-volume data.  Do this beforehand to check accessibility
;  852          0849  2 ! of all volumes in the set.  Read the file header into HDR_BUFFER.
;  853          0850  2 !
;  854          0851  2 INCR RVN FROM 2 TO .VOLUME_COUNT DO
;  855          0852  2     BEGIN
;  856          0853  2     CH$FILL(0, FIB$C_LENGTH, FIB);
;  857          0854  2     FIB[FIB$L_ACCTL] = ACCTL[.RVN-1] = FIB$M_WRITE OR FIB$M_NORECORD;
;  858          0855  2     FIB[FIB$W_FID_NUM] = FID$C_INDEXF;
;  859          0856  2     FIB[FIB$W_FID_SEQ] = FID$C_INDEXF;
;  860          0857  2     FIB[FIB$W_FID_RVN] = .RVN;
;  861        P 0858  2     STATUS = $QIOW(
;  862        P 0859  2         FUNC=IO$_ACCESS OR IO$M_ACCESS,
;  863        P 0860  2         CHAN=.CHANNEL,
;  864        P 0861  2         IOSB=IOSB,
;  865        P 0862  2         P1=FIB_DESC,
;  866          0863  2         P5=HDR_ATR_DESC);
;  867          0864  2     IF .STATUS THEN STATUS = .IOSB[0];
;  868          0865  2     IF .STATUS EQL SS$_WRITLCK
;  869          0866  2     THEN
;  870          0867  4         BEGIN
;  871          0868  4         CH$FILL(0, FIB$C_LENGTH, FIB);
;  872          0869  4         NO_WRITE = TRUE;
;  873          0870  4         FIB[FIB$L_ACCTL] = ACCTL[.RVN-1] = FIB$M_NORECORD;
;  874          0871  4         FIB[FIB$W_FID_NUM] = FID$C_INDEXF;
;  875          0872  4         FIB[FIB$W_FID_SEQ] = FID$C_INDEXF;
;  876          0873  4         FIB[FIB$W_FID_RVN] = .RVN;
;  877        P 0874  4         STATUS = $QIOW(
;  878        P 0875  4             FUNC=IO$_ACCESS OR IO$M_ACCESS,
;  879        P 0876  4             CHAN=.CHANNEL,
;  880        P 0877  4             IOSB=IOSB,
;  881        P 0878  4             P1=FIB_DESC,
;  882          0879  4             P5=HDR_ATR_DESC);
;  883          0880  4         IF .STATUS THEN STATUS = .IOSB[0];
;  884          0881  4         END;
;  885          0882  2     IF NOT .STATUS THEN SIGNAL(VERIFY$_OPENINDEX, 1, .RVN, .STATUS);
;  886          0883  2     READ_HOMEBLOCK(.RVN);
;  887          0884  2     INIT_VOL_DATA(.RVN, .ACCTL[.RVN-1]);
;  888          0885  2     END;
;  889          0886  2
;  890          0887  2
```

```
;  891          0888  2 ! If one or more volumes are write locked, cancel /REPAIR.
;  892          0889  2 !
;  893          0890  2 IF .NO_WRITE
;  894          0891  2 THEN
;  895          0892  2     IF TESTBITSC(QUAL[QUAL_REPA])
;  896          0893  2     THEN
;  897          0894  2         SIGNAL(VERIFY$_NOREPAIR);
;  898          0895  2
;  899          0896  2
;  900          0897  2 ! Initialize for quota processing.
;  901          0898  2 !
;  902          0899  2 QT[QT_LINK] = 0;
;  903          0900  2 QT[QT_COUNT] = QT_MAXCOUNT;
;  904          0901  2
;  905          0902  2
;  906          0903  2 IF .STRUCTURE_LEVEL EQL 2
;  907          0904  2 THEN
;  908          0905  2     BEGIN
;  909          0906  2
;  910          0907  2     ! Access the quota file.  If this fails, skip quota
;  911          0908  2     ! computation.
;  912          0909  2     !
;  913          0910  2     CH$FILL(0, FIB$C_LENGTH, FIB);
;  914          0911  2     FIB[FIB$L_ACCTL] = .ACCTL[0];
;  915          0912  2     FIB[FIB$W_DID_NUM] = FID$C_MFD;
;  916          0913  2     FIB[FIB$W_DID_SEQ] = FID$C_MFD;
;  917          0914  2     FIB[FIB$W_DID_RVN] = 1;
;  918        P 0915  2     STATUS = $QIOW(
;  919        P 0916  2         FUNC=IO$_ACCESS OR IO$M_ACCESS,
;  920        P 0917  2         CHAN=.CHANNEL,
;  921        P 0918  2         IOSB=IOSB,
;  922        P 0919  2         P1=FIB_DESC,
;  923        P 0920  2         P2=QFI_DESC,
;  924          0921  2         P5=FAT_ATR_DESC);
;  925          0922  2     IF .STATUS THEN STATUS = .IOSB[0];
;  926          0923  2
;  927          0924  2     IF .STATUS
;  928          0925  2     THEN
;  929          0926  2         IF
;  930          0927  2             NOT .UCHAR[FCH$V_CONTIG] OR
;  931          0928  2             .RECATTR[FAT$B_RTYPE] NEQ FAT$C_FIXED OR
;  932          0929  2             .RECATTR[FAT$B_RATTR[B] NEQ 0 OR
;  933          0930  2             .RECATTR[FAT$W_RSIZE] NEQ DQF$C_LENGTH
;  934          0931  2         THEN
;  935          0932  4             BEGIN
;  936        P 0933  4             $QIOW(
;  937        P 0934  4                 FUNC=IO$_DEACCESS,
;  938          0935  4                 CHAN=.CHANNEL);
;  939          0936  4             STATUS = SS$_BADQFILE;
;  940          0937  4             END;
;  941          0938  2
;  942          0939  2     IF NOT .STATUS
;  943          0940  2     THEN
;  944          0941  2         SIGNAL(VERIFY$_OPENQUOTA, 0, .STATUS)
;  945          0942  2     ELSE
;  946          0943  4         BEGIN
;  947          0944  4
```

```
;  948     0945  4          ! Mark quota processing enabled.
;  949     0946  4          !
;  950     0947  4          QUOTA_ACTIVE = TRUE;
;  951     0948  4
;  952     0949  4
;  953     0950  4          ! Loop to read the quota file.
;  954     0951  4          !
;  955     0952  4          INCR VBN FROM 1 TO ROT(.RECATTR[FAT$L_EFBLK], 16) - 1 BY INDEX_BUF_COUNT DO
;  956     0953  5              BEGIN
;  957     0954  5              LOCAL
;  958     0955  5                  THIS_BLOCKS;    ! Count of blocks to read on current iteration
;  959     0956  5
;  960     0957  5
;  961     0958  5              ! Compute number of blocks to read this time.
;  962     0959  5              !
;  963     0960  5              THIS_BLOCKS = MINU(
;  964     0961  5                  INDEX_BUF_COUNT,
;  965     0962  5                  ROT(.RECATTR[FAT$L_EFBLK], 16) - .VBN);
;  966     0963  5
;  967     0964  5
;  968     0965  5              ! Read the blocks.  If this fails, re-execute the read one block at
;  969     0966  5              ! a time noting the blocks that fail.  The buffer for each failed
;  970     0967  5              ! block is set to zero to effectively ignore that block.
;  971     0968  5              !
;  972   P 0969  5              STATUS = $QIOW(
;  973   P 0970  5                  FUNC=IO$_READVBLK,
;  974   P 0971  5                  CHAN=.CHANNEL,
;  975   P 0972  5                  IOSB=IOSB,
;  976   P 0973  5                  P1=BUFFER,
;  977   P 0974  5                  P2=.THIS_BLOCKS * 512,
;  978     0975  5                  P3=.VBN);
;  979     0976  5              IF .STATUS THEN STATUS = .IOSB[0];
;  980     0977  5              IF NOT .STATUS
;  981     0978  5              THEN
;  982     0979  6                  BEGIN
;  983     0980  6                  INCR XVBN FROM 0 TO .THIS_BLOCKS-1 DO
;  984     0981  7                      BEGIN
;  985   P 0982  7                      STATUS = $QIOW(
;  986   P 0983  7                          FUNC=IO$_READVBLK,
;  987   P 0984  7                          CHAN=.CHANNEL,
;  988   P 0985  7                          IOSB=IOSB,
;  989   P 0986  7                          P1=BUFFER + .XVBN * 512,
;  990   P 0987  7                          P2=512,
;  991     0988  7                          P3=.VBN + .XVBN);
;  992     0989  7                      IF .STATUS THEN STATUS = .IOSB[0];
;  993     0990  7                      IF NOT .STATUS
;  994     0991  7                      THEN
;  995     0992  8                          BEGIN
;  996     0993  8                          SIGNAL(VERIFY$_READQUOTA, 1, .VBN + .XVBN, .STATUS);
;  997     0994  8                          CH$FILL(0, 512, BUFFER + .XVBN * 512);
;  998     0995  7                          END;
;  999     0996  6                      END;
; 1000     0997  5                  END;
; 1001     0998  5
; 1002     0999  5
; 1003     1000  5              ! Loop to process each quota entry.
; 1004     1001  5              !
```

```
;  1005     1002  5            INCRA ENTRY FROM BUFFER TO BUFFER + .THIS_BLOCKS*512 - DQF$C_LENGTH BY DQF$C_LENGTH DO
;  1006     1003  6                BEGIN
;  1007     1004  6                MAP
;  1008     1005  6                    ENTRY:      REF BBLOCK;
;  1009     1006  6
;  1010     1007  6                IF .ENTRY[DQF$V_ACTIVE]
;  1011     1008  6                THEN
;  1012     1009  7                    BEGIN
;  1013     1010  7                    IF .ENTRY[DQF$L_UIC] EQL 0  ! Default record
;  1014     1011  7                    THEN
;  1015     1012  8                        BEGIN
;  1016     1013  8                        DEFAULT_QUOTA = .ENTRY[DQF$L_PERMQUOTA];
;  1017     1014  8                        DEFAULT_OVERDRAFT = .ENTRY[DQF$L_OVERDRAFT];
;  1018     1015  8                        END
;  1019     1016  7                    ELSE
;  1020     1017  8                        BEGIN
;  1021     1018  8                        LAST_UIC = .ENTRY[DQF$L_UIC];
;  1022     1019  8                        COUNT_QUOTA(.ENTRY[DQF$L_UIC], .ENTRY[DQF$L_USAGE], 0);
;  1023     1020  7                        END;
;  1024     1021  6                    END;
;  1025     1022  5                END;
;  1026     1023  4            END;
;  1027     1024  4
;  1028     1025  4
;  1029     1026  4        ! Deaccess the quota file.
;  1030     1027  4        !
;  1031   P 1028  4        $QIOW(
;  1032   P 1029  4            FUNC=IO$_DEACCESS,
;  1033     1030  4            CHAN=.CHANNEL);
;  1034     1031  3        END;
;  1035     1032  2    END;
;  1036     1033  2
;  1037     1034  2
;  1038     1035  2 ! Get the current time.  Also, convert it to ODS-1 format.
;  1039     1036  2 !
;  1040     1037  2 $GETTIM(TIMADR=CURRENT_TIME);
;  1041     1038  2 $ASCTIM(TIMBUF=UPLIT(23, BUFFER), TIMADR=CURRENT_TIME);
;  1042     1039  2 CURRENT_TIME_1[0,0,16,0] = .BUFFER[0,0,16,0];    ! Output DD
;  1043     1040  2 CURRENT_TIME_1[2,0,24,0] = .BUFFER[3,0,24,0];    ! Output MMM
;  1044     1041  2 CURRENT_TIME_1[5,0,16,0] = .BUFFER[9,0,16,0];    ! Output YY
;  1045     1042  2 CURRENT_TIME_1[7,0,16,0] = .BUFFER[12,0,16,0];   ! Output HH
;  1046     1043  2 CURRENT_TIME_1[9,0,16,0] = .BUFFER[15,0,16,0];   ! Output MM
;  1047     1044  2 CURRENT_TIME_1[11,0,16,0] = .BUFFER[18,0,16,0];  ! Output SS
;  1048     1045  2 IF .CURRENT_TIME_1[0,0,8,0] EQL %C' ' THEN CURRENT_TIME_1[0,0,8,0] = %C'0';
;  1049     1046  2
;  1050     1047  2
;  1051     1048  2 ! Scan the index files.
;  1052     1049  2 !
;  1053     1050  2 SCAN_INDEX();
;  1054     1051  2
;  1055     1052  2
;  1056     1053  2 ! Check for allocation errors.
;  1057     1054  2 !
;  1058     1055  2 INCR RVN FROM 1 TO .VOLUME_COUNT DO
;  1059     1056  3     BEGIN
;  1060     1057  3     LOCAL
;  1061     1058  3         SMAP:           REF BITVECTOR;          ! Pointer to old storage bitmap
```

```
;  1062          1059  3
;  1063          1060
;  1064          1061          ! Access bitmap file.
;  1065          1062          !
;  1066          1063          CH$FILL(0, FIB$C_LENGTH, FIB);
;  1067          1064          FIB[FIB$L_ACCTL] = .ACCTL[.RVN-1];
;  1068          1065          FIB[FIB$W_FID_NUM] = FID$C_BITMAP;
;  1069          1066          FIB[FIB$W_FID_SEQ] = FID$C_BITMAP;
;  1070          1067          FIB[FIB$W_FID_RVN] = .RVN;
;  1071        P 1068          STATUS = $QIOW(
;  1072        P 1069              FUNC=IO$_ACCESS OR IO$M_ACCESS,
;  1073        P 1070              CHAN=.CHANNEL,
;  1074        P 1071              IOSB=IOSB,
;  1075          1072              P1=FIB_DESC);
;  1076          1073          IF .STATUS THEN STATUS = .IOSB[0];
;  1077          1074          IF NOT .STATUS
;  1078          1075          THEN
;  1079          1076              SIGNAL(VERIFY$_OPENBITMAP, 1, .RVN, .STATUS);
;  1080          1077
;  1081          1078
;  1082          1079          ! Allocate space for copy of old storage bitmap.
;  1083          1080          !
;  1084          1081          STATUS = LIB$GET_VM(%REF(.SMAP_SIZE[.RVN-1] * 512), SMAP);
;  1085          1082          IF NOT .STATUS THEN SIGNAL(VERIFY$_ALLOCMEM, 0, .STATUS);
;  1086          1083
;  1087          1084
;  1088          1085          ! Read old storage bitmap.
;  1089          1086          !
;  1090          1087          INCR VBN FROM 0 TO .SMAP_SIZE[.RVN-1] - 1 BY 127 DO
;  1091          1088  4           BEGIN
;  1092          1089  4           LOCAL
;  1093          1090  4               THIS_BLOCKS;        ! Count of blocks to read on current iteration
;  1094          1091  4
;  1095          1092  4
;  1096          1093  4           ! Compute number of blocks to read this time.
;  1097          1094  4           !
;  1098          1095  4           THIS_BLOCKS = MINU(
;  1099          1096  4               127,
;  1100          1097  4               .SMAP_SIZE[.RVN-1] - .VBN);
;  1101          1098  4
;  1102          1099  4
;  1103          1100  4           ! Read the blocks.  If this fails, re-execute the read one block
;  1104          1101  4           ! at a time noting the blocks that fail.
;  1105          1102  4           !
;  1106        P 1103  4           STATUS = $QIOW(
;  1107        P 1104  4               FUNC=IO$_READVBLK,
;  1108        P 1105  4               CHAN=.CHANNEL,
;  1109        P 1106  4               IOSB=IOSB,
;  1110        P 1107  4               P1=.SMAP + .VBN * 512,
;  1111        P 1108  4               P2=.THIS_BLOCKS * 512,
;  1112          1109  4               P3=2 + .VBN);
;  1113          1110  4           IF .STATUS THEN STATUS = .IOSB[0];
;  1114          1111  4           IF NOT .STATUS
;  1115          1112  4           THEN
;  1116          1113  5               BEGIN
;  1117          1114  5               INCR XVBN FROM 0 TO .THIS_BLOCKS-1 DO
;  1118          1115  6                   BEGIN
```

```
;  1119     P 1116  6                  STATUS = $QIOW(
;  1120     P 1117  6                      FUNC=IO$_READVBLK,
;  1121     P 1118  6                      CHAN=.CHANNEL,
;  1122     P 1119  6                      IOSB=IOSB,
;  1123     P 1120  6                      P1=.SMAP + .VBN * 512 + .XVBN * 512,
;  1124     P 1121  6                      P2=512,
;  1125       1122  6                      P3=2 + .VBN + .XVBN);
;  1126       1123  6                  IF .STATUS THEN STATUS = .IOSB[0];
;  1127       1124  6                  IF NOT .STATUS
;  1128       1125  6                  THEN
;  1129       1126  6                      SIGNAL(
;  1130       1127  6                          VERIFY$_READSBMAP,
;  1131       1128  6                          2,
;  1132       1129  6                          2 + .VBN + .XVBN,
;  1133       1130  6                          .RVN,
;  1134       1131  6                          .STATUS);
;  1135       1132  5                  END;
;  1136       1133  4              END;
;  1137       1134  3          END;
;  1138       1135  3
;  1139       1136  3
;  1140       1137  3          ! Match "valid" storage bitmap against old, reporting discrepancies.
;  1141       1138  3          !
;  1142       1139  3          INCR J FROM 0 TO .SMAP_SIZE[.RVN-1]*512*8-1 DO
;  1143       1140  4              BEGIN
;  1144       1141  4              IF .SMAP[.J] NEQ .BITVECTOR[.VSMAP[.RVN-1], .J]
;  1145       1142  4              THEN
;  1146       1143  5                  BEGIN
;  1147       1144  5                  LOCAL
;  1148       1145  5                      KK;
;  1149       1146  5
;  1150       1147  5                  INCR K FROM .J TO .SMAP_SIZE[.RVN-1]*512*8-1 DO
;  1151       1148  6                      BEGIN
;  1152       1149  6                      IF
;  1153       1150  6                          .SMAP[.K] EQL .BITVECTOR[.VSMAP[.RVN-1], .K] OR
;  1154       1151  6                          .BITVECTOR[.VSMAP[.RVN-1], .K] NEQ .BITVECTOR[.VSMAP[.RVN-1], .J]
;  1155       1152  6                      THEN
;  1156       1153  6                          EXITLOOP;
;  1157       1154  6                      KK = .K;
;  1158       1155  5                      END;
;  1159       1156  5
;  1160       1157  5                  SIGNAL(
;  1161       1158  6                      (IF .SMAP[.J]
;  1162       1159  6                          THEN VERIFY$_ALLOCSET
;  1163       1160  5                          ELSE VERIFY$_ALLOCCLR),
;  1164       1161  5                      3,
;  1165       1162  5                      .J * .CLUSTER_FACTOR[.RVN-1],
;  1166       1163  5                      (.KK + 1) * .CLUSTER_FACTOR[.RVN-1] - 1,
;  1167       1164  5                      .RVN);
;  1168       1165  5
;  1169       1166  5                  J = .KK;
;  1170       1167  4                  END;
;  1171       1168  3              END;
;  1172       1169  3
;  1173       1170  3
;  1174       1171  3          ! Match "valid" storage bitmap against new, reporting discrepancies.  These
;  1175       1172  3          ! arise from lost extension headers.
```

```
;  1176     1173  3     !
;  1177     1174  3     INCR J FROM 0 TO .SMAP_SIZE[.RVN-1]*512*8-1 DO
;  1178     1175  4         BEGIN
;  1179     1176  4         IF .BITVECTOR[.NSMAP[.RVN-1], .J] NEQ .BITVECTOR[.VSMAP[.RVN-1], .J]
;  1180     1177  4         THEN
;  1181     1178  5             BEGIN
;  1182     1179  5             LOCAL
;  1183     1180  5                 KK;
;  1184     1181  5
;  1185     1182  5             INCR K FROM .J TO .SMAP_SIZE[.RVN-1]*512*8-1 DO
;  1186     1183  6                 BEGIN
;  1187     1184  6                 IF
;  1188     1185  6                     .BITVECTOR[.NSMAP[.RVN-1], .K] EQL .BITVECTOR[.VSMAP[.RVN-1], .K] OR
;  1189     1186  6                     .BITVECTOR[.VSMAP[.RVN-1], .K] NEQ .BITVECTOR[.VSMAP[.RVN-1], .J]
;  1190     1187  6                 THEN
;  1191     1188  6                     EXITLOOP;
;  1192     1189  6                 KK = .K;
;  1193     1190  5                 END;
;  1194     1191  5
;  1195     1192  5             SIGNAL(
;  1196     1193  5                 VERIFY$_ALLOCEXT,
;  1197     1194  5                 3,
;  1198     1195  5                 .J * .CLUSTER_FACTOR[.RVN-1],
;  1199     1196  5                 (.KK + 1) * .CLUSTER_FACTOR[.RVN-1] - 1,
;  1200     1197  5                 .RVN);
;  1201     1198  5
;  1202     1199  5             J = .KK;
;  1203     1200  4             END;
;  1204     1201  3         END;
;  1205     1202  3
;  1206     1203  3
;  1207     1204  3     ! Free space for copy of old storage bitmap.
;  1208     1205  3     !
;  1209     1206  3     STATUS = LIB$FREE_VM(%REF(.SMAP_SIZE[.RVN-1] * 512), SMAP);
;  1210     1207  3     IF NOT .STATUS THEN SIGNAL(VERIFY$_FREEMEM, 0, .STATUS);
;  1211     1208  3
;  1212     1209  3
;  1213     1210  3     ! Write new storage bitmap.
;  1214     1211  3     !
;  1215     1212  3     IF DO_REPAIR(NO_CONFIRM)
;  1216     1213  3     THEN
;  1217     1214  4         BEGIN
;  1218     1215  4         INCR VBN FROM 0 TO .SMAP_SIZE[.RVN-1] - 1 BY 127 DO
;  1219     1216  5             BEGIN
;  1220     1217  5             LOCAL
;  1221     1218  5                 THIS_BLOCKS;    ! Count of blocks to write on current iteration
;  1222     1219  5
;  1223     1220  5
;  1224     1221  5             ! Compute number of blocks to write this time.
;  1225     1222  5             !
;  1226     1223  5             THIS_BLOCKS = MINU(
;  1227     1224  5                 127,
;  1228     1225  5                 .SMAP_SIZE[.RVN-1] - .VBN);
;  1229     1226  5
;  1230     1227  5
;  1231     1228  5             ! Write the blocks.  If this fails, re-execute the write one block
;  1232     1229  5             ! at a time noting the blocks that fail.
```

```
;  1233        1230  5              !
;  1234      P 1231  5              STATUS = $QIOW(
;  1235      P 1232  5                  FUNC=IO$_WRITEVBLK,
;  1236      P 1233  5                  CHAN=.CHANNEL,
;  1237      P 1234  5                  IOSB=IOSB,
;  1238      P 1235  5                  P1=.NSMAP[.RVN-1] + .VBN * 512,
;  1239      P 1236  5                  P2=.THIS_BLOCKS * 512,
;  1240        1237  5                  P3=2 + .VBN);
;  1241        1238  5              IF .STATUS THEN STATUS = .IOSB[0];
;  1242        1239  5              IF NOT .STATUS
;  1243        1240  5              THEN
;  1244        1241  6                  BEGIN
;  1245        1242  6                  INCR XVBN FROM 0 TO .THIS_BLOCKS-1 DO
;  1246        1243  7                      BEGIN
;  1247      P 1244  7                      STATUS = $QIOW(
;  1248      P 1245  7                          FUNC=IO$_WRITEVBLK,
;  1249      P 1246  7                          CHAN=.CHANNEL,
;  1250      P 1247  7                          IOSB=IOSB,
;  1251      P 1248  7                          P1=.NSMAP[.RVN-1] + .VBN * 512 + .XVBN * 512,
;  1252      P 1249  7                          P2=512,
;  1253        1250  7                          P3=2 + .VBN + .XVBN);
;  1254        1251  7                      IF .STATUS THEN STATUS = .IOSB[0];
;  1255        1252  7                      IF NOT .STATUS
;  1256        1253  7                      THEN
;  1257        1254  7                          SIGNAL(
;  1258        1255  7                              VERIFY$_WRITESBMAP,
;  1259        1256  7                              2,
;  1260        1257  7                              2 + .VBN + .XVBN,
;  1261        1258  7                              .RVN,
;  1262        1259  7                              .STATUS);
;  1263        1260  6                      END;
;  1264        1261  5                  END;
;  1265        1262  4              END;
;  1266        1263                  END;
;  1267        1264
;  1268        1265
;  1269        1266                  ! Deaccess bitmap file.
;  1270        1267                  !
;  1271      P 1268                  $QIOW(
;  1272      P 1269                      FUNC=IO$_DEACCESS,
;  1273        1270                      CHAN=.CHANNEL);
;  1274        1271                  END;
;  1275        1272
;  1276        1273
;  1277        1274  2 ! If required, rescan the index files to locate multiple allocation.
;  1278        1275  2 !
;  1279        1276  2 IF .DUAL_ALLOC_FOUND
;  1280        1277  2 THEN
;  1281        1278  2     BEGIN
;  1282        1279  2     DUAL_ALLOC_PASS = TRUE;
;  1283        1280  2     SCAN_INDEX();
;  1284        1281  2     END;
;  1285        1282  2
;  1286        1283  2
;  1287        1284  2 ! Scan the extension header bitmap looking for lost extension headers.
;  1288        1285  2 !
;  1289        1286  2 INCR RVN FROM 1 TO .VOLUME_COUNT DO
```

```
;  1290     1287  3      BEGIN
;  1291     1288  3      INCR NUM FROM 1 TO .MAXFILIDX[.RVN-1]+1 DO
;  1292     1289  4          BEGIN
;  1293     1290  4          IF
;  1294     1291  4              .BITVECTOR[.IMAP[.RVN-1], .NUM-1] AND
;  1295     1292  4              NOT .BITVECTOR[.LOSTMAP[.RVN-1], .NUM-1] AND
;  1296     1293  4              NOT .BITVECTOR[.EXTMAP[.RVN-1], .NUM-1]
;  1297     1294  4          THEN
;  1298     1295  5              BEGIN
;  1299     1296  5              LOCAL
;  1300     1297  5                  FILE_ID:        BBLOCK[FID$C_LENGTH];   ! Local copy of file ID
;  1301     1298  5
;  1302     1299  5              FILE_ID[FID$W_NUM] = .NUM;
;  1303     1300  5              FILE_ID[FID$B_NMX] =  NUM<16,8>;
;  1304     1301  5              FILE_ID[FID$W_SEQ] =  ECTOR[.SEQMAP[.RVN-1], .NUM-1 ;,WORD];
;  1305     1302  5              FILE_ID[FID$B_RVN] = .RVN;
;  1306     1303  5              HEADER_ERROR(VERIFY$_LOSTEXTHDR, FILE_ID, 0);
;  1307     1304  5              IF DO_REPAIR(NO_CONFIRM)
;  1308     1305  5              THEN
;  1309     1306  5                  IF READ_HEADER(FILE_ID, BUFFER_2)
;  1310     1307  5                  THEN
;  1311     1308  5                      DELETE_HEADER(FILE_ID, BUFFER_2);
;  1312     1309  4              END;
;  1313     1310  3          END;
;  1314     1311  2      END;
;  1315     1312  2
;  1316     1313  2
;  1317     1314  2  ! Rewrite the index file bitmaps.
;  1318     1315  2  !
;  1319     1316  2  IF DO_REPAIR(NO_CONFIRM)
;  1320     1317  2  THEN
;  1321     1318  2      INCR RVN FROM 1 TO .VOLUME_COUNT DO
;  1322     1319  3          BEGIN
;  1323     1320  3
;  1324     1321  3          ! Access the index file.
;  1325     1322  3          !
;  1326     1323  3          CH$FILL(0, FIB$C_LENGTH, FIB);
;  1327     1324  3          FIB[FIB$L_ACCTL] = FIB$M_WRITE OR FIB$M_NORECORD;
;  1328     1325  3          FIB[FIB$W_FID_NUM] = FID$C_INDEXF;
;  1329     1326  3          FIB[FIB$W_FID_SEQ] = FID$C_INDEXF;
;  1330     1327  3          FIB[FIB$W_FID_RVN] = .RVN;
;  1331  P  1328  3          STATUS = $QIOW(
;  1332  P  1329  3              FUNC=IO$_ACCESS OR IO$M_ACCESS,
;  1333  P  1330  3              CHAN=.CHANNEL,
;  1334  P  1331  3              IOSB=IOSB,
;  1335     1332  3              P1=FIB_DESC);
;  1336     1333  3          IF .STATUS THEN STATUS = .IOSB[0];
;  1337     1334  3          IF NOT .STATUS THEN SIGNAL(VERIFY$_OPENINDEX, 1, .RVN, .STATUS);
;  1338     1335  3
;  1339     1336  3
;  1340     1337  3          ! Write new index file bitmap.
;  1341     1338  3          !
;  1342     1339  3          INCR VBN FROM 0 TO .IMAP_SIZE[.RVN-1] - 1 BY 127 DO
;  1343     1340  4              BEGIN
;  1344     1341  4              LOCAL
;  1345     1342  4                  THIS_BLOCKS;    ! Count of blocks to read on current iteration
;  1346     1343  4
```

```
;  1347          1344  4
;  1348          1345  4              ! Compute number of blocks to write this time.
;  1349          1346  4              !
;  1350          1347  4              THIS_BLOCKS = MINU(
;  1351          1348  4                  127,
;  1352          1349  4                  .IMAP_SIZE[.RVN-1] - .VBN);
;  1353          1350  4
;  1354          1351  4
;  1355          1352  4              ! Write the blocks.  If this fails, re-execute the write one block
;  1356          1353  4              ! at a time noting the blocks that fail.
;  1357          1354  4              !
;  1358        P 1355  4              STATUS = $QIOW(
;  1359        P 1356  4                  FUNC=IO$_WRITEVBLK,
;  1360        P 1357  4                  CHAN=.CHANNEL,
;  1361        P 1358  4                  IOSB=IOSB,
;  1362        P 1359  4                  P1=.IMAP[.RVN-1] + .VBN * 512,
;  1363        P 1360  4                  P2=.THIS_BLOCKS * 512,
;  1364          1361  4                  P3=.BITMAP_OFFSET[.RVN-1] + .VBN);
;  1365          1362  4              IF .STATUS THEN STATUS = .IOSB[0];
;  1366          1363  4              IF NOT .STATUS
;  1367          1364  4              THEN
;  1368          1365  5                  BEGIN
;  1369          1366  5                  INCR XVBN FROM 0 TO .THIS_BLOCKS-1 DO
;  1370          1367  6                      BEGIN
;  1371        P 1368  6                      STATUS = $QIOW(
;  1372        P 1369  6                          FUNC=IO$_WRITEVBLK,
;  1373        P 1370  6                          CHAN=.CHANNEL,
;  1374        P 1371  6                          IOSB=IOSB,
;  1375        P 1372  6                          P1=.IMAP[.RVN-1] + .VBN * 512 + .XVBN * 512,
;  1376        P 1373  6                          P2=512,
;  1377          1374  6                          P3=.BITMAP_OFFSET[.RVN-1] + .VBN + .XVBN);
;  1378          1375  6                      IF .STATUS THEN STATUS = .IOSB[0];
;  1379          1376  6                      IF NOT .STATUS
;  1380          1377  6                      THEN
;  1381          1378  6                          SIGNAL(
;  1382          1379  6                              VERIFY$_WRITEIBMAP,
;  1383          1380  6                              2,
;  1384          1381  6                              .BITMAP_OFFSET[.RVN-1] + .VBN + .XVBN,
;  1385          1382  6                              .RVN,
;  1386          1383  6                              .STATUS);
;  1387          1384  5                      END;
;  1388          1385  4                  END;
;  1389          1386  3              END;
;  1390          1387  3
;  1391          1388  3
;  1392          1389  3              ! Deaccess the index file.
;  1393          1390  3              !
;  1394        P 1391  3              $QIOW(
;  1395        P 1392  3                  FUNC=IO$_DEACCESS,
;  1396          1393  3                  CHAN=.CHANNEL);
;  1397          1394  2              END;
;  1398          1395  2
;  1399          1396  2
;  1400          1397  2  ! Scan all directories on all volumes.
;  1401          1398  2  !
;  1402          1399  2  INCR RVN FROM 1 TO .VOLUME_COUNT DO
;  1403          1400  3      BEGIN
```

```
;  1404     1401  3      DIR_SCAN(.RVN);
;  1405     1402  3      END;
;  1406     1403  3
;  1407     1404  3
;  1408     1405  2  ! Scan the lost file bitmap looking for lost files.
;  1409     1406  2  !
;  1410     1407  2  IF .DIRECTORY_ERROR THEN SIGNAL(VERIFY$_LOSTSCAN);
;  1411     1408  2
;  1412     1409  2
;  1413     1410  2  INCR RVN FROM 1 TO .VOLUME_COUNT DO
;  1414     1411  3      BEGIN
;  1415     1412  3      INCR NUM FROM 1 TO .MAXFILIDX[.RVN-1]+1 DO
;  1416     1413  4          BEGIN
;  1417     1414  4          IF .BITVECTOR[.LOSTMAP[.RVN-1], .NUM-1]
;  1418     1415  4          THEN
;  1419     1416  5              BEGIN
;  1420     1417  5              LOCAL
;  1421     1418  5                  FILE_ID:        BBLOCK[FID$C_LENGTH];   ! Local copy of file ID
;  1422     1419  5
;  1423     1420  5
;  1424     1421  5              ! Get the file ID.
;  1425     1422  5              !
;  1426     1423  5              FILE_ID[FID$W_NUM] = .NUM;
;  1427     1424  5              FILE_ID[FID$B_NMX] = .NUM<16,8>;
;  1428     1425  5              FILE_ID[FID$W_SEQ] = .VECTOR[.SEQMAP[.RVN-1], .NUM-1 ;,WORD];
;  1429     1426  5              FILE_ID[FID$B_RVN] = .RVN;
;  1430     1427  5
;  1431     1428  5
;  1432     1429  5              ! Generate usage file entry for the lost file if requested.
;  1433     1430  5              !
;  1434     1431  5              IF .QUAL[QUAL_USAG] THEN IF READ_HEADER(FILE_ID, BUFFER_2)
;  1435     1432  5              THEN
;  1436     1433  6                  BEGIN
;  1437     1434  6                  LOCAL
;  1438     1435  6                      FILENAME:           VECTOR[FI2$S_FILENAME + FI2$S_FILENAMEXT + 1, BYTE],
;  1439     1436  6                      LENGTH,
;  1440     1437  6                      IDENT_AREA:         REF BBLOCK;     ! Pointer to ident area
;  1441     1438  6
;  1442     1439  6
;  1443     1440  6                  IDENT_AREA = BUFFER_2 + .BUFFER_2[FH2$B_IDOFFSET]*2;
;  1444     1441  6                  IF .STRUCTURE_LEVEL EQL 2
;  1445     1442  6                  THEN
;  1446     1443  7                      BEGIN
;  1447     1444  7                      CH$COPY(
;  1448     1445  7                          FI2$S_FILENAME, IDENT_AREA[FI2$T_FILENAME],
;  1449     1446  7                          %C' ',
;  1450     1447  7                          FI2$S_FILENAME + FI2$S_FILENAMEXT + 1, FILENAME);
;  1451     1448  7
;  1452     1449  7                      IF (.BUFFER_2[FH2$B_MPOFFSET] - .BUFFER_2[FH2$B_IDOFFSET]) * 2
;  1453     1450  7                      GEQU $BYTEOFFSET(FI2$T_FILENAMEXT) + FI2$S_FILENAMEXT
;  1454     1451  7                      THEN
;  1455     1452  7                          CH$MOVE(
;  1456     1453  7                              FI2$S_FILENAMEXT,
;  1457     1454  7                              IDENT_AREA[FI2$T_FILENAMEXT],
;  1458     1455  7                              FILENAME[FI2$S_FILENAME]);
;  1459     1456  7
;  1460     1457  7                      LENGTH =
```

```
;  1461          1458  7                    CH$FIND_CH(FI2$S_FILENAME + FI2$S_FILENAMEXT + 1, FILENAME, %C' ')
;  1462          1459  7                    - FILENAME;
;  1463          1460  7                END
;  1464          1461  6            ELSE
;  1465          1462  7                BEGIN
;  1466          1463  7                LENGTH = MAKE_STRING(
;  1467          1464  7                    IDENT_AREA[FI1$W_FILENAME] - $BYTEOFFSET(NMB$W_NAME),
;  1468          1465  7                    FILENAME);
;  1469          1466  6                END;
;  1470          1467  6
;  1471          1468  6
;  1472          1469  6            USAGE_BUFFER[USG$B_TYPE] = USG$K_FILE;
;  1473          1470  6            USAGE_BUFFER[USG$L_FILEOWNER] = .VECTOR[.OWNER[.RVN-1], .NUM-1];
;  1474          1471  6            USAGE_BUFFER[USG$L_ALLOCATED] = .VECTOR[.ALLOCATION[.RVN-1], .NUM-1];
;  1475          1472  6            USAGE_BUFFER[USG$L_USED] = .VECTOR[.USAGE[.RVN-1], .NUM-1];
;  1476          1473  6            USAGE_BUFFER[USG$W_DIR_LEN] = 2;
;  1477        P 1474  6            $FAO(
;  1478        P 1475  6                $DESCRIPTOR('[]!AF'),
;  1479        P 1476  6                USAGE_BUFFER[USG$W_SPEC_LEN],
;  1480        P 1477  6                UPLIT(
;  1481        P 1478  6                    %ALLOCATION(USAGE_BUFFER) - $BYTEOFFSET(USG$T_FILESPEC),
;  1482        P 1479  6                    USAGE_BUFFER[USG$T_FILESPEC]),
;  1483          1480  6                .LENGTH, FILENAME);
;  1484          1481  6            USAGE_RAB[RAB$W_RSZ] = $BYTEOFFSET(USG$T_FILESPEC) + .USAGE_BUFFER[USG$W_SPEC_LEN];
;  1485          1482  7            IF NOT $PUT(RAB=USAGE_RAB)
;  1486          1483  6            THEN
;  1487          1484  6                FILE_ERROR(
;  1488          1485  6                    VERIFY$_FACILITY + SHR$_WRITEERR + STS$K_SEVERE,
;  1489          1486  6                    USAGE_FAB,
;  1490          1487  6                    .USAGE_RAB[RAB$L_STS], .USAGE_RAB[RAB$L_STV]);
;  1491          1488  5            END;
;  1492          1489  5
;  1493          1490  5
;  1494          1491  5            ! Report the lost file if no errors occurred in the directory scan.
;  1495          1492  5            !
;  1496          1493  5            IF NOT .DIRECTORY_ERROR
;  1497          1494  5            THEN
;  1498          1495  6                BEGIN
;  1499          1496  6                HEADER_ERROR(VERIFY$_LOSTHEADER, FILE_ID, 0);
;  1500          1497  7                IF (STATUS = DO_REPAIR(ALLOW_DELETE))
;  1501          1498  6                THEN
;  1502          1499  6                    ENTER_WORK(
;  1503          1500  7                        (IF .STATUS<1,1>
;  1504          1501  7                            THEN WRK_K_DELETE
;  1505          1502  6                            ELSE WRK_K_ENTER),
;  1506          1503  6                        FILE_ID);
;  1507          1504  5                END;
;  1508          1505  4            END;
;  1509          1506  3        END;
;  1510          1507  2    END;
;  1511          1508  2
;  1512          1509  2
;  1513          1510  2! Scan the quota table looking for quota errors.
;  1514          1511  2!
;  1515          1512  2IF .QUOTA_ACTIVE
;  1516          1513  2THEN
;  1517          1514  3    BEGIN
```

```
;  1518          1515  3     LOCAL
;  1519          1516  3         M,                          ! True if modify quota, false if add quota
;  1520          1517  3         T:      REF BBLOCK,         ! Pointer to table segment
;  1521          1518  3         E:      REF BBLOCK;         ! Pointer to table entry
;  1522          1519  3
;  1523          1520  3
;  1524          1521  3     IF DO_REPAIR(NO_CONFIRM)
;  1525          1522  3     THEN
;  1526          1523  4         BEGIN
;  1527          1524  4
;  1528          1525  4         ! Enable the quota file.  This is required to execute the
;  1529          1526  4         ! modify-quota function.
;  1530          1527  4         !
;  1531          1528  4         CH$FILL(0, FIB$C_LENGTH, FIB);
;  1532          1529  4         FIB[FIB$W_DID_NUM] = FID$C_MFD;
;  1533          1530  4         FIB[FIB$W_DID_SEQ] = FID$C_MFD;
;  1534          1531  4         FIB[FIB$W_DID_RVN] = 1;
;  1535          1532  4         FIB[FIB$W_CNTRLFUNC] = FIB$C_ENA_QUOTA;
;  1536        P 1533  4         STATUS = $QIOW(
;  1537        P 1534  4             FUNC=IO$_ACPCONTROL,
;  1538        P 1535  4             CHAN=.CHANNEL,
;  1539        P 1536  4             IOSB=IOSB,
;  1540        P 1537  4             P1=FIB_DESC,
;  1541          1538  4             P2=QFI_DESC);
;  1542          1539  4         IF .STATUS THEN STATUS = .IOSB[0];
;  1543          1540  4         IF .STATUS NEQ SS$_QFACTIVE
;  1544          1541  4         THEN
;  1545          1542  5             BEGIN
;  1546          1543  5             IF NOT .STATUS THEN SIGNAL(VERIFY$_ENAQUOTA, 0, .STATUS);
;  1547          1544  5             QUOTA_DISABLE = TRUE;
;  1548          1545  4             END;
;  1549          1546  3         END;
;  1550          1547  3
;  1551          1548  3
;  1552          1549  3     M = TRUE;
;  1553          1550  3     IF .LAST_UIC EQL 0 THEN M = FALSE;
;  1554          1551  3     T = .QT[QT_LINK];
;  1555          1552  3
;  1556          1553  3
;  1557          1554  3     WHILE .T NEQ 0 DO
;  1558          1555  4         BEGIN
;  1559          1556  4         E = .T + QT_S_HDR;
;  1560          1557  4         INCR J FROM 0 TO .T[QT_COUNT]-1 DO
;  1561          1558  5             BEGIN
;  1562          1559  5             IF .E[QT_QUO_USED] NEQ .E[QT_IDX_USED]
;  1563          1560  5             THEN
;  1564          1561  6                 BEGIN
;  1565          1562  6
;  1566          1563  6                 ! Quota file and computed value disagree.
;  1567          1564  6                 !
;  1568          1565  6                 SIGNAL(
;  1569          1566  6                     VERIFY$_INCQUOTA,
;  1570          1567  6                     3,
;  1571          1568  6                     .E[QT_QUO_USED],
;  1572          1569  6                     .E[QT_IDX_USED],
;  1573          1570  6                     .E[QT_UIC]);
;  1574          1571  6                 IF DO_REPAIR()
```

```
;  1575     1572  6              THEN
;  1576     1573  6                  IF .M
;  1577     1574  6                  THEN
;  1578     1575  7                      BEGIN
;  1579     1576  7
;  1580     1577  7                      ! Modify existing quota file entry.
;  1581     1578  7                      !
;  1582     1579  7                      CH$FILL(0, FIB$C_LENGTH, FIB);
;  1583     1580  7                      FIB[FIB$W_CNTRLFUNC] = FIB$C_MOD_QUOTA;
;  1584     1581  7                      FIB[FIB$L_CNTRLVAL] = FIB$M_MOD_OSE;
;  1585     1582  7                      DQF[DQF$L_UIC] = .E[QT_UIC];
;  1586     1583  7                      DQF[DQF$L_USAGE] = .E[QT_IDX_USED];
;  1587   P 1584  7                      STATUS = $QIOW(
;  1588   P 1585  7                          FUNC=IO$_ACPCONTROL,
;  1589   P 1586  7                          CHAN=.CHANNEL,
;  1590   P 1587  7                          IOSB=IOSB,
;  1591   P 1588  7                          P1=FIB_DESC,
;  1592     1589  7                          P2=DQF_DESC);
;  1593     1590  7                      IF .STATUS THEN STATUS = .IOSB[0];
;  1594     1591  7                      IF NOT .STATUS
;  1595     1592  7                      THEN
;  1596     1593  7                          SIGNAL(
;  1597     1594  7                              VERIFY$_MODQUOTA,
;  1598     1595  7                              1,
;  1599     1596  7                              .E[QT_UIC],
;  1600     1597  7                              .STATUS);
;  1601     1598  7                      END
;  1602     1599  6                  ELSE
;  1603     1600  7                      BEGIN
;  1604     1601  7
;  1605     1602  7                      ! Add new quota file entry, after volume is unlocked.
;  1606     1603  7                      !
;  1607     1604  7                      ENTER_WORK(WRK_K_ADDQUO, .E[QT_UIC], .E[QT_IDX_USED]);
;  1608     1605  6                      END;
;  1609     1606  5                  END;
;  1610     1607  5
;  1611     1608  5
;  1612     1609  5              IF .E[QT_UIC] EQL .LAST_UIC THEN M = FALSE;
;  1613     1610  5              E = .E + QT_S_ENT;
;  1614     1611  4              END;
;  1615     1612  4          T = .T[QT_LINK];
;  1616     1613  3          END;
;  1617     1614  2      END;
;  1618     1615  2
;  1619     1616  2
;  1620     1617  2  ! Unlock the volume set.
;  1621     1618  2  !
;  1622     1619  2  IF .QUAL[QUAL_REPA]
;  1623     1620  2  THEN
;  1624     1621  3      BEGIN
;  1625     1622  3      CH$FILL(0, FIB$C_LENGTH, FIB);
;  1626     1623  3      FIB[FIB$W_CNTRLFUNC] = FIB$C_UNLK_VOL;
;  1627   P 1624  3      STATUS = $QIOW(
;  1628   P 1625  3          FUNC=IO$_ACPCONTROL,
;  1629   P 1626  3          CHAN=.CHANNEL,
;  1630   P 1627  3          IOSB=IOSB,
;  1631     1628  3          P1=FIB_DESC);
```

```
;  1632     1629  3      IF .STATUS THEN STATUS = .IOSB[0];
;  1633     1630  3      IF NOT .STATUS THEN SIGNAL(VERIFY$_UNLKVOL, 0, .STATUS);
;  1634     1631  3      QUAL[QUAL_REPA] = FALSE;
;  1635     1632  3      END;
;  1636     1633  3
;  1637     1634  3
;  1638     1635  3  ! Do delayed repairs.
;  1639     1636  3  !
;  1640     1637  3  PROCESS_WORK();
;  1641     1638  3
;  1642     1639  3
;  1643     1640  3  ! Disable the quota file, if it was disabled before we began processing.
;  1644     1641  3  !
;  1645     1642  3  IF .QUOTA_DISABLE
;  1646     1643  3  THEN
;  1647     1644  3      BEGIN
;  1648     1645  3      CH$FILL(0, FIB$C_LENGTH, FIB);
;  1649     1646  3      FIB[FIB$W_CNTRLFUNC] = FIB$C_DSA_QUOTA;
;  1650   P 1647  3      STATUS = $QIOW(
;  1651   P 1648  3          FUNC=IO$_ACPCONTROL,
;  1652   P 1649  3          CHAN=.CHANNEL,
;  1653   P 1650  3          IOSB=IOSB,
;  1654     1651  3          P1=FIB_DESC);
;  1655     1652  3      IF .STATUS THEN STATUS = .IOSB[0];
;  1656     1653  3      IF NOT .STATUS
;  1657     1654  3      THEN
;  1658     1655  3          SIGNAL(VERIFY$_DSAQUOTA, 0, .STATUS);
;  1659     1656  3      QUOTA_DISABLE = FALSE;
;  1660     1657  3      END;
;  1661     1658  3
;  1662     1659  3
;  1663     1660  2  ! If necessary, deaccess a file that is accessed on the alternate channel.
;  1664     1661  2  ! Then, deassign the channels to the device.
;  1665     1662  2  !
;  1666     1663  2  ACCESS_INDEX_2(0);
;  1667     1664  2  $DASSGN(CHAN=.CHANNEL);
;  1668     1665  2  $DASSGN(CHAN=.CHANNEL_2);
;  1669     1666  2
;  1670     1667  2
;  1671     1668  2  ! Close the output listing file.
;  1672     1669  2  !
;  1673     1670  2  IF .QUAL[QUAL_LIST]
;  1674     1671  2  THEN
;  1675     1672  3      IF NOT $CLOSE(FAB=LIST_FAB)
;  1676     1673  3      THEN
;  1677     1674  3          FILE_ERROR(
;  1678     1675  3              VERIFY$_FACILITY^16 + SHR$_CLOSEOUT + STS$K_SEVERE,
;  1679     1676  3              LIST_FAB,
;  1680     1677  3              .LIST_FAB[FAB$L_STS], .LIST_FAB[FAB$L_STV]);
;  1681     1678  3
;  1682     1679  3
;  1683     1680  2  ! Close the output usage file.
;  1684     1681  2  !
;  1685     1682  2  IF .QUAL[QUAL_USAG]
;  1686     1683  2  THEN
;  1687     1684  3      IF NOT $CLOSE(FAB=USAGE_FAB)
;  1688     1685  2      THEN
```

```
;  1689          1686  2          FILE_ERROR(
;  1690          1687  2              VERIFY$_FACILITY^16 + SHR$_CLOSEOUT + STS$K_SEVERE,
;  1691          1688  2              USAGE_FAB,
;  1692          1689  2              .USAGE_FAB[FAB$L_STS], .USAGE_FAB[FAB$L_STV]);
;  1693          1690  2
;  1694          1691  2
;  1695          1692  2      ! Return to operating system.
;  1696          1693  2      !
;  1697          1694  2      $SS$_NORMAL
;  1698          1695  1      END;
```

```
                                        .TITLE  VERIFY Main module
                                        .IDENT  \V04-000\

                                        .PSECT  DATA,NOEXE,2

                              00000 QUAL:       .BLKB   4
                              00004 QUAL_DEV_DESC:
                                        .BLKB   8
                              0000C QUAL_LIST_DESC:
                                        .BLKB   8
                              00014 QUAL_USAG_DESC:
                                        .BLKB   8
                              0001C LIST_FAB:
                                        .BLKB   80
                              0006C LIST_RAB:
                                        .BLKB   68
                              000B0 LIST_NAM:
                                        .BLKB   96
                              00110 LIST_RSA:
                                        .BLKB   255
                              0020F             .BLKB   1
                              00210 LIST_DESC:
                                        .BLKB   8
                              00218 LIST_BUFFER:
                                        .BLKB   132
                              0029C USAGE_FAB:
                                        .BLKB   80
                              002EC USAGE_RAB:
                                        .BLKB   68
                              00330 USAGE_NAM:
                                        .BLKB   96
                              00390 USAGE_RSA:
                                        .BLKB   255
                              0048F             .BLKB   1
                              00490 USAGE_BUFFER:
                                        .BLKB   423
                              00637             .BLKB   1
                              00638 CHANNEL_TT:
                                        .BLKB   4
                              0063C CHANNEL:    .BLKB   4
                              00640 CHANNEL_2:
                                        .BLKB   4
                              00644 CHAN2_RVN:
                                        .BLKB   4
                              00648 TOTAL_SIZE:
```

```
                        .BLKB   4
0064C DUAL_ALLOC_FOUND:
                        .BLKB   4
00650 DUAL_ALLOC_PASS:
                        .BLKB   4
00654 DIRECTORY_ERROR:
                        .BLKB   4
00658 OLD_CTRL_MASK:
                        .BLKB   4
0065C EXIT_HAND_DESC:
                        .BLKB   20
00670 QT:               .BLKB   8
00678 QUOTA_ACTIVE:
                        .BLKB   4
0067C QUOTA_DISABLE:
                        .BLKB   4
00680 DEFAULT_QUOTA:
                        .BLKB   4
00684 DEFAULT_OVERDRAFT:
                        .BLKB   4
00688 LAST_UIC:
                        .BLKB   4
0068C DQF:              .BLKB   32
006AC BUFFER:           .BLKB   32768
086AC BUFFER_2:
                        .BLKB   512
088AC FIB:              .BLKB   64
088EC RECATTR:          .BLKB   32
0890C UCHAR:            .BLKB   4
08910 DIR:              .BLKB   320
08A50 DIR_DESC:
                        .BLKB   8
08A58 DIR_FID:          .BLKB   6
08A5E                   .BLKB   2
08A60 LOST_DIR_FID:
                        .BLKB   6
08A66                   .BLKB   2
08A68 IOSB:             .BLKB   8
08A70 VOLUME_COUNT:
                        .BLKB   4
08A74 STRUCTURE_LEVEL:
                        .BLKB   4
08A78 HOMEVBN:          .BLKB   4
08A7C WORK_LIST:
                        .BLKB   8
08A84 CURRENT_TIME:
                        .BLKB   8
08A8C CURRENT_TIME_1:
                        .BLKB   13
08A99                   .BLKB   3
08A9C DEVICE_DESC:
                        .BLKB   8
08AA4 DEVICE_NAME:
                        .BLKB   16
08AB4 DEVICE_CHAR:
                        .BLKB   16
08AC4 PER_VOLUME_BEG:
```

```
                                                08AC4                   .BLKB   0
                                                08AC4 ACCTL:            .BLKB   4
                                                08AC8 IMAP_SIZE:
                                                                        .BLKB   4
                                                08ACC MAXFILIDX:
                                                                        .BLKB   4
                                                08AD0 IMAP:             .BLKB   4
                                                08AD4 DIRMAP:           .BLKB   4
                                                08AD8 SEQMAP:           .BLKB   4
                                                08ADC BACKMAP:          .BLKB   4
                                                08AE0 LOSTMAP:          .BLKB   4
                                                08AE4 EXTMAP:           .BLKB   4
                                                08AE8 OWNER:            .BLKB   4
                                                08AEC ALLOCATION:
                                                                        .BLKB   4
                                                08AF0 USAGE:            .BLKB   4
                                                08AF4 SMAP_SIZE:
                                                                        .BLKB   4
                                                08AF8 VSMAP:            .BLKB   4
                                                08AFC NSMAP:            .BLKB   4
                                                08B00 MULTSMAP:
                                                                        .BLKB   4
                                                08B04 CLUSTER_FACTOR:
                                                                        .BLKB   4
                                                08B08 HEADER_OFFSET:
                                                                        .BLKB   4
                                                08B0C BITMAP_OFFSET:
                                                                        .BLKB   4
                                                08B10 EOF:              .BLKB   4
                                                08B14 PER_VOLUME_END:
                                                                        .BLKB   0
31  3B  52  49  44  2E  54  53  4F  4C  53  59  53  08B14 LOST_NAME:
                                                                        .ASCII  \SYSLOST.DIR;1\                 ;
                                                08B21                   .BLKB   3
        31  3B  53  59  53  2E  41  54  4F  55  51  08B24 QFI_NAME:
                                                                        .ASCII  \QUOTA.SYS;1\                   ;

                                                                        .PSECT  CODE,NOWRT,2

                        30  30  30  30  30  30  00000 P.AAB:            .ASCII  \000000\                        ;
                                                00006                   .BLKB   2
                                  00000006      00008 P.AAA:            .LONG   6                               ;
                                  00000000'     0000C                   .ADDRESS P.AAB                          ;
                                0004  0020      00010 P.AAC:            .WORD   32, 4                           ;
                                  00000000'     00014                   .ADDRESS RECATTR                        ;
                                0003  0004      00018                   .WORD   4, 3                            ;
                                  00000000'     0001C                   .ADDRESS UCHAR                          ;
                                  00000000      00020                   .LONG   0                               ;
                                000A  0200      00024 P.AAD:            .WORD   512, 10                         ;
                                  00000000'     00028                   .ADDRESS HDR_BUFFER                     ;
                                  00000000      0002C                   .LONG   0                               ;
                                  00000040      00030 P.AAE:            .LONG   64                              ;
                                  00000000'     00034                   .ADDRESS FIB                            ;
                                  00000020      00038 P.AAF:            .LONG   32                              ;
                                  00000000'     0003C                   .ADDRESS DQF                            ;
                                  0000000D      00040 P.AAG:            .LONG   13                              ;
                                  00000000'     00044                   .ADDRESS LOST_NAME                      ;
```

```
                                00000000B  00048 P.AAH:  .LONG   11
                                00000000'  0004C        .ADDRESS QFI_NAME
                          45 43 49 56 45 44  00050 P.AAJ:  .ASCII  \DEVICE\
                                           00056        .BLKB   2
                                00000006.  00058 P.AAI:  .LONG   6
                                00000000'  0005C        .ADDRESS P.AAJ
                              0003  0004.   00060 P.AAK:  .WORD   4, 3
                                00000000'  00064        .ADDRESS DEVICE_CHAR
                                00000000   00068        .LONG   0
                              0032  0010.   0006C        .WORD   16, 50
                          00000000' 00000000' 00070      .ADDRESS DEVICE_NAME, DEVICE_DESC
                                00000000   00078        .LONG   0
                             54 53 49 4C   0007C P.AAM:  .ASCII  \LIST\
                                00000004.  00080 P.AAL:  .LONG   4
                                00000000'  00084        .ADDRESS P.AAM
                             54 53 49 4C   00088 P.AAO:  .ASCII  \LIST\
                                00000004.  0008C P.AAN:  .LONG   4
                                00000000'  00090        .ADDRESS P.AAO
                          45 47 41 53 55   00094 P.AAQ:  .ASCII  \USAGE\
                                           00099        .BLKB   3
                                00000005.  0009C P.AAP:  .LONG   5
                                00000000'  000A0        .ADDRESS P.AAQ
                          45 47 41 53 55   000A4 P.AAS:  .ASCII  \USAGE\
                                           000A9        .BLKB   3
                                00000005.  000AC P.AAR:  .LONG   5
                                00000000'  000B0        .ADDRESS P.AAS
                       4D 52 49 46 4E 4F 43  000B4 P.AAU:  .ASCII  \CONFIRM\
                                           000BB        .BLKB   1
                                00000007.  000BC P.AAT:  .LONG   7
                                00000000'  000C0        .ADDRESS P.AAU
                    4B 43 45 48 43 5F 44 41 45 52  000C4 P.AAW:  .ASCII  \READ_CHECK\
                                           000CE        .BLKB   2
                                0000000A.  000D0 P.AAV:  .LONG   10
                                00000000'  000D4        .ADDRESS P.AAW
                             52 49 41 50 45 52  000D8 P.AAY:  .ASCII  \REPAIR\
                                           000DE        .BLKB   2
                                00000006.  000E0 P.AAX:  .LONG   6
                                00000000'  000E4        .ADDRESS P.AAY
                    53 49 4C 2E 59 46 49 52 45 56  000E8 P.AAZ:  .ASCII  \VERIFY.LIS\
                 3A 54 55 50 54 55 4F 24 53 59 53  000F2 P.ABA:  .ASCII  \SYS$OUTPUT:\
64 6E 69 20 66 6F 20 67 6E 69 74 73 69 4C 21  000FD P.ABB:  .ASCII  \!Listing of index file on !AS!/!%D\
21 53 41 21 20 6E 6F 20 65 6C 69 66 20 78 65  0010C
                                44 25 21 2F  0011B
                       54 41 44 2E 45 47 41 53 55  0011F P.ABC:  .ASCII  \USAGE.DAT\
                 44 4E 41 4D 4D 4F 43 24 53 59 53  00128 P.ABE:  .ASCII  \SYS$COMMAND\
                                           00133        .BLKB   1
                                0000000B.  00134 P.ABD:  .LONG   11
                                00000000'  00138        .ADDRESS P.ABE
                                00000017.  0013C P.ABF:  .LONG   23
                                00000000'  00140        .ADDRESS BUFFER
                             46 41 21 5D 5B  00144 P.ABH:  .ASCII  \[]!AF\
                                           00149        .BLKB   3
                                00000005.  0014C P.ABG:  .LONG   5
                                00000000'  00150        .ADDRESS P.ABH
                                00000196.  00154 P.ABI:  .LONG   406
                                00000000'  00158        .ADDRESS USAGE_BUFFER+17
```

```
HDR_BUFFER=         BUFFER+512
HDR_BUFFER_2=       BUFFER+1024
MFD_DESC=           P.AAA
FAT_ATR_DESC=       P.AAC
HDR_ATR_DESC=       P.AAD
FIB_DESC=           P.AAE
DQF_DESC=           P.AAF
LOST_DESC=          P.AAG
QFI_DESC=           P.AAH
$RMS_PTR=           LIST_FAB
$RMS_PTR=           LIST_NAM
$RMS_PTR=           LIST_FAB
$RMS_PTR=           LIST_RAB
$RMS_PTR=           LIST_NAM
$RMS_PTR=           USAGE_FAB
$RMS_PTR=           USAGE_RAB
$RMS_PTR=           USAGE_NAM
                .EXTRN  CHECKSUM, CHECKSUM2
                .EXTRN  LEFT_ONE, MAKE_STRING
                .EXTRN  CLI$GET_VALUE, CLI$PRESENT
                .EXTRN  LIB$DISABLE_CTRL
                .EXTRN  LIB$ENABLE_CTRL
                .EXTRN  LIB$FREE_VM, LIB$GET_COMMAND
                .EXTRN  LIB$GET_VM, LIB$SCOPY_R_DX
                .EXTRN  LIB$SIGNAL, VERIFY$_FACILITY
                .EXTRN  VERIFY$_ABORT, VERIFY$_ADDQUOTA
                .EXTRN  VERIFY$_ALLOCCLR
                .EXTRN  VERIFY$_ALLOCEXT
                .EXTRN  VERIFY$_ALLOCMEM
                .EXTRN  VERIFY$_ALLOCSET
                .EXTRN  VERIFY$_ALTIHDBAD
                .EXTRN  VERIFY$_ASSIGN, VERIFY$_BACKLINK
                .EXTRN  VERIFY$_BADBITMAP
                .EXTRN  VERIFY$_BADDIR, VERIFY$_BADDIRENT
                .EXTRN  VERIFY$_BADEFBLK
                .EXTRN  VERIFY$_BADHEADER
                .EXTRN  VERIFY$_BADHIBLK
                .EXTRN  VERIFY$_BBLHEADER
                .EXTRN  VERIFY$_CHKALTHOME
                .EXTRN  VERIFY$_CHKPRIHOME
                .EXTRN  VERIFY$_CHKSCB, VERIFY$_CREATELOST
                .EXTRN  VERIFY$_DELETE, VERIFY$_DELHEADER
                .EXTRN  VERIFY$_DIRNAME
                .EXTRN  VERIFY$_DSAQUOTA
                .EXTRN  VERIFY$_ENAQUOTA
                .EXTRN  VERIFY$_ENTERLOST
                .EXTRN  VERIFY$_FINDHOME
                .EXTRN  VERIFY$_FINDIHD
                .EXTRN  VERIFY$_FREEMEM
                .EXTRN  VERIFY$_FUTBAKDAT
                .EXTRN  VERIFY$_FUTCREDAT
                .EXTRN  VERIFY$_FUTREVDAT
                .EXTRN  VERIFY$_GETDVI, VERIFY$_INCQUOTA
                .EXTRN  VERIFY$_INVDEVICE
                .EXTRN  VERIFY$_INVEXTBACK
                .EXTRN  VERIFY$_INVEXTFID
                .EXTRN  VERIFY$_INVEXTHDR
```

```
                                                  .EXTRN  VERIFY$_LOCKHEADER
                                                  .EXTRN  VERIFY$_LOCKVOL
                                                  .EXTRN  VERIFY$_LOSTEXTHDR
                                                  .EXTRN  VERIFY$_LOSTHEADER
                                                  .EXTRN  VERIFY$_LOSTSCAN
                                                  .EXTRN  VERIFY$_MAPAREA
                                                  .EXTRN  VERIFY$_MAXVOLS
                                                  .EXTRN  VERIFY$_MODQUOTA
                                                  .EXTRN  VERIFY$_MULTALLOC
                                                  .EXTRN  VERIFY$_MULTEXTHDR
                                                  .EXTRN  VERIFY$_NOREPAIR
                                                  .EXTRN  VERIFY$_OPENBITMAP
                                                  .EXTRN  VERIFY$_OPENDIR
                                                  .EXTRN  VERIFY$_OPENFILE
                                                  .EXTRN  VERIFY$_OPENINDEX
                                                  .EXTRN  VERIFY$_OPENQUOTA
                                                  .EXTRN  VERIFY$_PRIIHDBAD
                                                  .EXTRN  VERIFY$_READBOOT
                                                  .EXTRN  VERIFY$_READDIR
                                                  .EXTRN  VERIFY$_READFILE
                                                  .EXTRN  VERIFY$_READHEADER
                                                  .EXTRN  VERIFY$_READHOME
                                                  .EXTRN  VERIFY$_READIBMAP
                                                  .EXTRN  VERIFY$_READQUOTA
                                                  .EXTRN  VERIFY$_READSBMAP
                                                  .EXTRN  VERIFY$_READSCB
                                                  .EXTRN  VERIFY$_REMOVE, VERIFY$_UNLKVOL
                                                  .EXTRN  VERIFY$_WRITEHEADER
                                                  .EXTRN  VERIFY$_WRITEHOME
                                                  .EXTRN  VERIFY$_WRITEIBMAP
                                                  .EXTRN  VERIFY$_WRITESBMAP
                                                  .EXTRN  VERIFY$_WRITESCB
                                                  .EXTRN  VERIFY$_WRONGOWNER
                                                  .EXTRN  SYS$PARSE, SYS$GETDVI
                                                  .EXTRN  SYS$CREATE, SYS$CONNECT
                                                  .EXTRN  SYS$ASSIGN, SYS$QIOW
                                                  .EXTRN  SYS$DCLEXH, SYS$GETTIM
                                                  .EXTRN  SYS$PUT, SYS$ASCTIM
                                                  .EXTRN  SYS$FAO, SYS$DASSGN
                                                  .EXTRN  SYS$CLOSE

                                   0FFC 00000 VERIFY: .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11         ; 0489
                  5E        94   AE   9E 00002          MOVAB   -108(SP), SP                                 ;
         00000000'  EF            02   90 00006          MOVB    #2, QUAL_DEV_DESC+3                          ; 0525
                    00000000'    EF   9F 0000D          PUSHAB  QUAL_DEV_DESC                                ; 0526
                         FEE5    CF   9F 00013          PUSHAB  P_AAT                                        ;
         00000000G  00            02   FB 00017          CALLS   #2, CLI$GET_VALUE                            ;
0050  8F        00  6E            00   2C 0001E          MOVC5   #0, (SP), #0, #80, $RMS_PTR                  ; 0536
                    00000000'    EF      00025                                                               ;
         00000000'  EF   5003    8F   B0 0002A          MOVW    #20483, $RMS_PTR                             ;
         00000000'  EF            02   90 00033          MOVB    #2, $RMS_PTR+22                              ;
         00000000'  EF            02   90 0003A          MOVB    #2, $RMS_PTR+31                              ;
         00000000'  EF 00000000'  EF   9E 00041          MOVAB   LIST_NAM, $RMS_PTR+40                        ;
         00000000'  EF 00000000'  EF   D0 0004C          MOVL    QUAL_DEV_DESC+4, $RMS_PTR+44                 ;
         00000000'  EF 00000000'  EF   90 00057          MOVB    QUAL_DEV_DESC, $RMS_PTR+52                   ;
0060  8F        00  6E            00   2C 00062          MOVC5   #0, (SP), #0, #96, $RMS_PTR                  ; 0539
                    00000000'    EF      00069                                                               ;
```

```
                00000000'  EF     6002   8F  B0 0006E          MOVW    #24578, $RMS_PTR
                00000000'  EF            01  8E 00077          MNEGB   #1, $RMS_PTR+10
                00000000'  EF 00000000'  EF  9E 0007E          MOVAB   LIST_RSA, $RMS_PTR+12
                           00000000'     EF  9F 00089          PUSHAB  LIST_FAB                        ; 0540
                00000000G  00            01  FB 0008F          CALLS   #1, SYS$PARSE
                           00000000'     EF  95 00096          TSTB    LIST_NAM+11                     ; 0542
                                         0D  13 0009C          BEQL    1$
                00060047   8F 00000000'  EF  D3 0009E          BITL    LIST_NAM+52, #393287            ; 0549
                                         15  13 000A9          BEQL    2$
                           00000000'     EF  9F 000AB 1$:      PUSHAB  QUAL_DEV_DESC                   ; 0551
                                         01  DD 000B1          PUSHL   #1
                           00000000G     8F  DD 000B3          PUSHL   #VERIFY$_INVDEVICE
                00000000G  00            03  FB 000B9          CALLS   #3, LIB$SIGNAL
                           00000000'     EF  9F 000C0 2$:      PUSHAB  QUAL_DEV_DESC                   ; 0556
                           00000000'     EF  9F 000C6          PUSHAB  LIST_RSA
                      0C   AE 00000000'  EF  9A 000CC          MOVZBL  LIST_NAM+57, 12(SP)             ; 0557
                                0C   AE  9F 000D4          PUSHAB  12(SP)
                00000000G  00            03  FB 000D7          CALLS   #3, LIB$SCOPY_R_DX
                           00000000'     EF  D4 000DE          CLRL    DEVICE_DESC                     ; 0565
                00000000'  EF 00000000'  EF  9E 000E4          MOVAB   DEVICE_NAME, DEVICE_DESC+4      ; 0566
                                         7E  7C 000EF          CLRQ    -(SP)                           ; 0574
                                         7E  7C 000F1          CLRQ    -(SP)
                                   FE0D  CF  9F 000F3          PUSHAB  P.AAK
                           00000000'     EF  9F 000F7          PUSHAB  QUAL_DEV_DESC
                                         7E  7C 000FD          CLRQ    -(SP)
                00000000G  00            08  FB 000FF          CALLS   #8, SYS$GETDVI
                           59            50  D0 00106          MOVL    R0, STATUS
                           13            59  E8 00109          BLBS    STATUS, 3$                      ; 0575
                                         59  DD 0010C          PUSHL   STATUS                          ; 0577
                                         01  DD 0010E          PUSHL   #1
                                         01  DD 00110          PUSHL   #1
                           00000000G     8F  DD 00112          PUSHL   #VERIFY$_GETDVI
                      00000000G  00            04  FB 00118          CALLS   #4, LIB$SIGNAL
                   15 00000000'  EF            04  E0 0011F 3$:      BBS     #4, DEVICE_CHAR+3, 4$   ; 0580
                                 00000000'     EF  9F 00127          PUSHAB  QUAL_DEV_DESC           ; 0582
                                               01  DD 0012D          PUSHL   #1
                                 00000000G     8F  DD 0012F          PUSHL   #VERIFY$_INVDEVICE
                      00000000G  00            03  FB 00135          CALLS   #3, LIB$SIGNAL
                      00000000'  EF            02  90 0013C 4$:      MOVB    #2, QUAL_LIST_DESC+3    ; 0587
                                         FDDD  CF  9F 00143          PUSHAB  P.AAL                   ; 0588
                      00000000G  00            01  FB 00147          CALLS   #1, CLI$PRESENT
                                 18            50  E9 0014E          BLBC    R0, 5$
                      00000000'  EF            02  88 00151          BISB2   #2, QUAL                ; 0591
                                 00000000'     EF  9F 00158          PUSHAB  QUAL_LIST_DESC          ; 0592
                                         FDCE  CF  9F 0015E          PUSHAB  P.AAN
                      00000000G  00            02  FB 00162          CALLS   #2, CLI$GET_VALUE
                      00000000'  EF            02  90 00169 5$:      MOVB    #2, QUAL_USAG_DESC+3    ; 0598
                                         FDCC  CF  9F 00170          PUSHAB  P.AAP                   ; 0599
                      00000000G  00            01  FB 00174          CALLS   #1, CLI$PRESENT
                                 18            50  E9 0017B          BLBC    R0, 6$
                      00000000'  EF            10  88 0017E          BISB2   #16, QUAL               ; 0602
                                 00000000'     EF  9F 00185          PUSHAB  QUAL_USAG_DESC          ; 0603
                                         FDC1  CF  9F 0018B          PUSHAB  P.AAR
                      00000000G  00            02  FB 0018F          CALLS   #2, CLI$GET_VALUE
                                         FDC6  CF  9F 00196 6$:      PUSHAB  P.AAT                   ; 0609
                      00000000G  00            01  FB 0019A          CALLS   #1, CLI$PRESENT
00000000'  EF               01                50  F0 001A1          INSV    R0, #0, #1, QUAL
```

```
                                        FDC6    CF  9F 001AA          PUSHAB  P.AAV                                   ; 0610
                 00000000G  00                  01  FB 001AE          CALLS   #1, CLI$PRESENT                         ;
00000000'  EF         01    02                  50  F0 001B5          INSV    R0, #2, #1, QUAL                        ;
                                        FDC2    CF  9F 001BE          PUSHAB  P.AAX                                   ; 0611
                 00000000G  00                  01  FB 001C2          CALLS   #1, CLI$PRESENT                         ;
00000000'  EF         01    03                  50  F0 001C9          INSV    R0, #3, #1, QUAL                        ;
                      03 00000000'  EF                01  E0 001D2          BBS     #1, QUAL, 7$                    ; 0616
                                              0168  31 001DA          BRW     11$                                     ;
    0050   8F         00          6E                00  2C 001DD 7$:      MOVC5   #0, (SP), #0, #80, $RMS_PTR             ; 0629
                                        00000000'  EF       001E4                                                       ;
                      00000000'  EF       5003    8F  B0 001E9          MOVW    #20483, $RMS_PTR                        ;
                      00000000'  EF         40    8F  9A 001F2          MOVZBL  #64, $RMS_PTR+4                         ;
                      00000000'  EF                 01  90 001FA          MOVB    #1, $RMS_PTR+22                         ;
                      00000000'  EF       0200    8F  B0 00201          MOVW    #512, $RMS_PTR+29                       ;
                      00000000'  EF                 02  90 0020A          MOVB    #2, $RMS_PTR+31                         ;
                      00000000'  EF  00000000'  EF  9E  00211          MOVAB   LIST_NAM, $RMS_PTR+40                   ;
                      00000000'  EF  00000000'  EF  D0  0021C          MOVL    QUAL_LIST_DESC+4, $RMS_PTR+44           ;
                      00000000'  EF    FD61     CF  9E  00227          MOVAB   P.AAZ, $RMS_PTR+48                      ;
                      00000000'  EF  00000000'  EF  90  00230          MOVB    QUAL_LIST_DESC, $RMS_PTR+52             ;
                      00000000'  EF                 0A  90  00238          MOVB    #10, $RMS_PTR+53                        ;
    0044   8F         00          6E                00  2C 00242          MOVC5   #0, (SP), #0, #68, $RMS_PTR             ; 0633
                                        00000000'  EF       00249                                                       ;
                      00000000'  EF       4401    8F  B0 0024E          MOVW    #17409, $RMS_PTR                        ;
                      00000000'  EF        0400   8F  3C 00257          MOVZWL  #1024, $RMS_PTR+4                       ;
                      00000000'  EF  00000000'  EF  9E  00260          MOVAB   LIST_BUFFER, $RMS_PTR+40                ;
                      00000000'  EF  00000000'  EF  9E  0026B          MOVAB   LIST_FAB, $RMS_PTR+60                   ;
    0060   8F         00          6E                00  2C 00276          MOVC5   #0, (SP), #0, #96, $RMS_PTR             ; 0638
                                        00000000'  EF       0027D                                                       ;
                      00000000'  EF       6002    8F  B0 00282          MOVW    #24578, $RMS_PTR                        ;
                      00000000'  EF                 01  8E  0028B          MNEGB   #1, $RMS_PTR+2                          ;
                      00000000'  EF  00000000'  EF  9E  00292          MOVAB   LIST_RSA, $RMS_PTR+4                    ;
                      00000000'  EF                 01  8E  0029D          MNEGB   #1, $RMS_PTR+10                         ;
                      00000000'  EF  00000000'  EF  9E  002A4          MOVAB   LIST_RSA, $RMS_PTR+12                   ;
                                     00000000'  EF  95  002AF          TSTB    LIST_FAB+52                             ; 0639
                                                10  12  002B5          BNEQ    8$                                      ;
                      00000000'  EF                 0B  90  002B7          MOVB    #11, LIST_FAB+52                        ; 0642
                      00000000'  EF         FCD4    CF  9E  002BE          MOVAB   P.ABA, LIST_FAB+44                      ; 0643
                      00000000'  EF           84    8F  9A  002C7 8$:      MOVZBL  #132, LIST_DESC                         ; 0645
                      00000000'  EF  00000000'  EF  9E  002CF          MOVAB   LIST_BUFFER, LIST_DESC+4                ; 0646
                                     00000000'  EF  9F  002DA          PUSHAB  LIST_FAB                                ; 0647
                 00000000G  00                  01  FB 002E0          CALLS   #1, SYS$CREATE                          ;
                            18                  50  E8 002E7          BLBS    R0, 9$                                  ;
                               7E  00000000'  EF  7D  002EA          MOVQ    LIST_FAB+8, -(SP)                       ; 0652
                                   00000000'  EF  9F  002F1          PUSHAB  LIST_FAB                                ; 0649
                                   00000000*  8F  DD  002F7          PUSHL   #<<<VERIFY$_FACILITY@16>+4256>+4>       ; 0650
                     0000V  CF                  04  FB 002FD          CALLS   #4, FILE_ERROR                          ;
                                   00000000'  EF  9F  00302 9$:       PUSHAB  LIST_RAB                                ; 0653
                 00000000G  00                  01  FB 00308          CALLS   #1, SYS$CONNECT                         ;
                            18                  50  E8 0030F          BLBS    R0, 10$                                 ;
                               7E  00000000'  EF  7D  00312          MOVQ    LIST_RAB+8, -(SP)                       ; 0658
                                   00000000'  EF  9F  00319          PUSHAB  LIST_FAB                                ; 0655
                                   00000000*  8F  DD  0031F          PUSHL   #<<<VERIFY$_FACILITY@16>+4256>+4>       ; 0656
                     0000V  CF                  04  FB 00325          CALLS   #4, FILE_ERROR                          ;
                                                7E  D4  0032A 10$:     CLRL    -(SP)                                   ; 0663
                                   00000000'  EF  9F  0032C          PUSHAB  QUAL_DEV_DESC                           ;
                                       FC6B   CF  9F  00332          PUSHAB  P.ABB                                   ;
                     0000V  CF                  03  FB 00336          CALLS   #3, FAO                                 ;
```

```
                        0000V  CF              00  FB 0033B          CALLS   #0, EOL                                          ;0664
                        0000V  CF              00  FB 00340          CALLS   #0, EOL                                          ;0665
                    03 00000000'  EF           04  E0 00345 11$:     BBS     #4, QUAL, 12$                                    ;0671
                                             0122  31 0034D          BRW     14$
    0050  8F        00             6E          00  2C 00350 12$:     MOVC5   #0, (SP), #0, #80, $RMS_PTR                      ;0684
                                 00000000'      EF  00357
                        00000000'  EF     5003  8F  B0 0035C          MOVW    #20483, $RMS_PTR
                        00000000'  EF       40  8F  9A 00365          MOVZBL  #64, $RMS_PTR+4
                        00000000'  EF           01  90 0036D          MOVB    #1, $RMS_PTR+22
                        00000000'  EF     0200  8F  B0 00374          MOVW    #512, $RMS_PTR+29
                        00000000'  EF           02  90 0037D          MOVB    #2, $RMS_PTR+31
                        00000000'  EF  00000000'  EF  9E 00384          MOVAB   USAGE_NAM, $RMS_PTR+40
                        00000000'  EF  00000000'  EF  D0 0038F          MOVL    QUAL_USAG_DESC+4, $RMS_PTR+44
                        00000000'  EF     FC25  CF  9E 0039A          MOVAB   P.ABC, $RMS_PTR+48
                        00000000'  EF  00000000'  EF  90 003A3          MOVB    QUAL_USAG_DESC, $RMS_PTR+52
                        00000000'  EF           09  90 003AE          MOVB    #9, $RMS_PTR+53
    0044  8F        00             6E          00  2C 003B5          MOVC5   #0, (SP), #0, #68, $RMS_PTR                      ;0688
                                 00000000'      EF  003BC
                        00000000'  EF     4401  8F  B0 003C1          MOVW    #17409, $RMS_PTR
                        00000000'  EF     0400  8F  3C 003CA          MOVZWL  #1024, $RMS_PTR+4
                        00000000'  EF  00000000'  EF  9E 003D3          MOVAB   USAGE_BUFFER, $RMS_PTR+40
                        00000000'  EF  00000000'  EF  9E 003DE          MOVAB   USAGE_FAB, $RMS_PTR+60
    0060  8F        00             6E          00  2C 003E9          MOVC5   #0, (SP), #0, #96, $RMS_PTR                      ;0693
                                 00000000'      EF  003F0
                        00000000'  EF     6002  8F  B0 003F5          MOVW    #24578, $RMS_PTR
                        00000000'  EF           01  8E 003FE          MNEGB   #1, $RMS_PTR+2
                        00000000'  EF  00000000'  EF  9E 00405          MOVAB   USAGE_RSA, $RMS_PTR+4
                        00000000'  EF           01  8E 00410          MNEGB   #1, $RMS_PTR+10
                        00000000'  EF  00000000'  EF  9E 00417          MOVAB   USAGE_RSA, $RMS_PTR+12
                                   00000000'      EF  9F 00422          PUSHAB  USAGE_FAB                                        ;0694
                    00000000G   00             01  FB 00428          CALLS   #1, SYS$CREATE
                                18             50  E8 0042F          BLBS    R0, 13$
                                7E  00000000'  EF  7D 00432          MOVQ    USAGE_FAB+8, -(SP)                               ;0699
                                    00000000'  EF  9F 00439          PUSHAB  USAGE_FAB                                        ;0696
                                    00000000*  8F  DD 0043F          PUSHL   #<<<VERIFY$_FACILITY&16>+4256>+4>                 ;0697
                        0000V   CF             04  FB 00445          CALLS   #4, FILE_ERROR
                                    00000000'  EF  9F 0044A 13$:     PUSHAB  USAGE_RAB                                        ;0700
                    00000000G   00             01  FB 00450          CALLS   #1, SYS$CONNECT
                                18             50  E8 00457          BLBS    R0, 14$
                                7E  00000000'  EF  7D 0045A          MOVQ    USAGE_RAB+8, -(SP)                               ;0705
                                    00000000'  EF  9F 00461          PUSHAB  USAGE_FAB                                        ;0702
                                    00000000*  8F  DD 00467          PUSHL   #<<<VERIFY$_FACILITY&16>+4256>+4>                 ;0703
                        0000V   CF             04  FB 0046D          CALLS   #4, FILE_ERROR
                                7E  7C  00472 14$:     CLRQ    -(SP)                                                            ;0711
                                    00000000'  EF  9F 00474          PUSHAB  CHANNEL
                                    00000000'  EF  9F 0047A          PUSHAB  QUAL_DEV_DESC
                    00000000G   00             04  FB 00480          CALLS   #4, SYS$ASSIGN
                                59             50  D0 00487          MOVL    R0, STATUS
                                17             59  E8 0048A          BLBS    STATUS, 15$                                      ;0712
                                59  DD  0048D          PUSHL   STATUS
                                    00000000'  EF  9F 0048F          PUSHAB  QUAL_DEV_DESC
                                    01  DD  00495          PUSHL   #1
                                    00000000G  8F  DD 00497          PUSHL   #VERIFY$_ASSIGN
                    00000000G   00             04  FB 0049D          CALLS   #4, LIB$SIGNAL
                                7E  7C  004A4 15$:     CLRQ    -(SP)                                                            ;0713
                                    00000000'  EF  9F 004A6          PUSHAB  CHANNEL_2
                                    00000000'  EF  9F 004AC          PUSHAB  QUAL_DEV_DESC
```

```
                00000000G  00              04  FB  004B2          CALLS   #4, SYS$ASSIGN                                          ;
                           59              50  D0  004B9          MOVL    R0, STATUS                                              ;
                           17              59  E8  004BC          BLBS    STATUS, 16$                                             ; 0714
                                           59  DD  004BF          PUSHL   STATUS                                                  ;
                                00000000'  EF  9F  004C1          PUSHAB  QUAL_DEV_DESC                                           ;
                                           01  DD  004C7          PUSHL   #1                                                      ;
                                00000000G  8F  DD  004C9          PUSHL   #VERIFY$_ASSIGN                                         ;
                00000000G  00              04  FB  004CF          CALLS   #4, LIB$SIGNAL                                          ;
                                           7E  7C  004D6  16$:     CLRQ    -(SP)                                                   ; 0720
                                00000000'  EF  9F  004D8          PUSHAB  CHANNEL_TT                                              ;
                                    FAF6   CF  9F  004DE          PUSHAB  P.ABD                                                   ;
                00000000G  00              04  FB  004E2          CALLS   #4, SYS$ASSIGN                                          ;
                           59              50  D0  004E9          MOVL    R0, STATUS                                              ;
                           65              59  E9  004EC          BLBC    STATUS, 17$                                             ; 0721
                                00000000'  EF  9F  004EF          PUSHAB  OLD_CTRL_MASK                                           ; 0723
                      08   AE  02000000    8F  D0  004F5          MOVL    #33554432, 8(SP)                                        ;
                               08          AE  9F  004FD          PUSHAB  8(SP)                                                   ;
                00000000G  00              02  FB  00500          CALLS   #2, LIB$DISABLE_CTRL                                    ;
                                           7E  7C  00507          CLRQ    -(SP)                                                   ; 0729
                           7E              03  7D  00509          MOVQ    #3, -(SP)                                               ;
                                           7E  D4  0050C          CLRL    -(SP)                                                   ;
                                    0000V  CF  9F  0050E          PUSHAB  CTRL_AST                                                ;
                                           7E  7C  00512          CLRQ    -(SP)                                                   ;
                                00000000'  EF  9F  00514          PUSHAB  IOSB                                                    ;
                           7E       0123   8F  3C  0051A          MOVZWL  #291, -(SP)                                             ;
                                00000000'  EF  DD  0051F          PUSHL   CHANNEL_TT                                              ;
                                           7E  D4  00525          CLRL    -(SP)                                                   ;
                00000000G  00              0C  FB  00527          CALLS   #12, SYS$QIOW                                           ;
                                           7E  7C  0052E          CLRQ    -(SP)                                                   ; 0735
                           7E              03  7D  00530          MOVQ    #3, -(SP)                                               ;
                                           7E  D4  00533          CLRL    -(SP)                                                   ;
                                    0000V  CF  9F  00535          PUSHAB  CTRL_AST                                                ;
                                           7E  7C  00539          CLRQ    -(SP)                                                   ;
                                00000000'  EF  9F  0053B          PUSHAB  IOSB                                                    ;
                           7E       A3     8F  9A  00541          MOVZBL  #163, -(SP)                                             ;
                                00000000'  EF  DD  00545          PUSHL   CHANNEL_TT                                              ;
                                           7E  D4  0054B          CLRL    -(SP)                                                   ;
                00000000G  00              0C  FB  0054D          CALLS   #12, SYS$QIOW                                           ;
                00000000'  EF       0000V  CF  9E  00554  17$:     MOVAB   EXIT_HANDLER, EXIT_HAND_DESC+4                          ; 0740
                00000000'  EF              01  D0  0055D          MOVL    #1, EXIT_HAND_DESC+8                                    ; 0741
                00000000'  EF  00000000'   EF  9E  00564          MOVAB   EXIT_HAND_DESC+16, EXIT_HAND_DESC+12                    ; 0742
                           00000000'       EF  9F  0056F          PUSHAB  EXIT_HAND_DESC                                          ; 0743
                00000000G  00              01  FB  00575          CALLS   #1, SYS$DCLEXH                                          ;
             5E 00000000'  EF              03  E1  0057C          BBC     #3, QUAL, 19$                                           ; 0745
0040    8F       00                       6E  00  2C  00584          MOVC5   #0, (SP), #0, #64, FIB                              ; 0750
                                00000000'  EF      0058B                                                                        ;
                00000000'  EF              07  B0  00590          MOVW    #7, FIB+22                                              ; 0751
                                           7E  7C  00597          CLRQ    -(SP)                                                   ; 0756
                                           7E  7C  00599          CLRQ    -(SP)                                                   ;
                                           7E  D4  0059B          CLRL    -(SP)                                                   ;
                                    F933   CF  9F  0059D          PUSHAB  FIB_DESC                                                ;
                                           7E  7C  005A1          CLRQ    -(SP)                                                   ;
                                00000000'  EF  9F  005A3          PUSHAB  IOSB                                                    ;
                                           38  DD  005A9          PUSHL   #56                                                     ;
                                00000000'  EF  DD  005AB          PUSHL   CHANNEL                                                 ;
                                           7E  D4  005B1          CLRL    -(SP)                                                   ;
                00000000G  00              0C  FB  005B3          CALLS   #12, SYS$QIOW                                           ;
```

```
                                59              50  D0 005BA          MOVL    R0, STATUS                                      ;
                                0A              59  E9 005BD          BLBC    STATUS, 18$                                     ; 0757
                                59 00000000'    EF  3C 005C0          MOVZWL  IOSB, STATUS                                    ;
                                18              59  E8 005C7          BLBS    STATUS, 19$                                     ; 0758
                                                59  DD 005CA 18$:     PUSHL   STATUS                                          ; 0761
                                                7E  D4 005CC          CLRL    -(SP)                                           ;
                                00000000G       8F  DD 005CE          PUSHL   #VERIFY$_LOCKVOL                                ;
                    00000000G   00              03  FB 005D4          CALLS   #3, LIB$SIGNAL                                  ;
                    00000000'   EF              08  8A 005DB          BICB2   #8, QUAL                                        ; 0762
0040    8F      00              6E              00  2C 005E2 19$:     MOVC5   #0, (SP), #0, #64, FIB                          ; 0769
                                00000000'       EF      005E9                                                               ;
                                                5A  D4 005EE          CLRL    NO_WRITE                                        ; 0770
                                58 00200100     8F  D0 005F0          MOVL    #2097408, ACCTL_0                               ; 0771
                    00000000'   EF 00200100     8F  D0 005F7          MOVL    #2097408, FIB                                   ;
                    00000000'   EF 00010001     8F  D0 00602          MOVL    #65537, FIB+4                                   ; 0772
                    00000000'   EF              01  B0 0060D          MOVW    #1, FIB+8                                       ; 0774
                                                7E  D4 00614          CLRL    -(SP)                                           ; 0780
                                    F8AE        CF  9F 00616          PUSHAB  HDR_ATR_DESC                                    ;
                                                7E  7C 0061A          CLRQ    -(SP)                                           ;
                                                7E  D4 0061C          CLRL    -(SP)                                           ;
                                    F8B2        CF  9F 0061E          PUSHAB  FIB_DESC                                        ;
                                                7E  7C 00622          CLRQ    -(SP)                                           ;
                                00000000'       EF  9F 00624          PUSHAB  IOSB                                            ;
                                7E      72      8F  9A 0062A          MOVZBL  #114, -(SP)                                     ;
                                00000000'       EF  DD 0062E          PUSHL   CHANNEL                                         ;
                                                7E  D4 00634          CLRL    -(SP)                                           ;
                    00000000G   00              0C  FB 00636          CALLS   #12, SYS$QIOW                                   ;
                                59              50  D0 0063D          MOVL    R0, STATUS                                      ;
                                07              59  E9 00640          BLBC    STATUS, 20$                                     ; 0781
                                59 00000000'    EF  3C 00643          MOVZWL  IOSB, STATUS                                    ;
                        0000025C    8F          59  D1 0064A 20$:     CMPL    STATUS, #604                                    ; 0782
                                                69  12 00651          BNEQ    21$                                             ;
0040    8F      00              6E              00  2C 00653          MOVC5   #0, (SP), #0, #64, FIB                          ; 0785
                                00000000'       EF      0065A                                                               ;
                                5A              01  D0 0065F          MOVL    #1, NO_WRITE                                    ; 0786
                                58 00200000     8F  D0 00662          MOVL    #2097152, ACCTL_0                               ; 0787
                    00000000'   EF 00200000     8F  D0 00669          MOVL    #2097152, FIB                                   ;
                    00000000'   EF 00010001     8F  D0 00674          MOVL    #65537, FIB+4                                   ; 0788
                    00000000'   EF              01  B0 0067F          MOVW    #1, FIB+8                                       ; 0790
                                                7E  D4 00686          CLRL    -(SP)                                           ; 0796
                                    F83C        CF  9F 00688          PUSHAB  HDR_ATR_DESC                                    ;
                                                7E  7C 0068C          CLRQ    -(SP)                                           ;
                                                7E  D4 0068E          CLRL    -(SP)                                           ;
                                    F840        CF  9F 00690          PUSHAB  FIB_DESC                                        ;
                                                7E  7C 00694          CLRQ    -(SP)                                           ;
                                00000000'       EF  9F 00696          PUSHAB  IOSB                                            ;
                                7E      72      8F  9A 0069C          MOVZBL  #114, -(SP)                                     ;
                                00000000'       EF  DD 006A0          PUSHL   CHANNEL                                         ;
                                                7E  D4 006A6          CLRL    -(SP)                                           ;
                    00000000G   00              0C  FB 006A8          CALLS   #12, SYS$QIOW                                   ;
                                59              50  D0 006AF          MOVL    R0, STATUS                                      ;
                                0A              59  E9 006B2          BLBC    STATUS, 22$                                     ; 0797
                                59 00000000'    EF  3C 006B5          MOVZWL  IOSB, STATUS                                    ;
                                13              59  E8 006BC 21$:     BLBS    STATUS, 23$                                     ; 0799
                                                59  DD 006BF 22$:     PUSHL   STATUS                                          ;
                                                01  DD 006C1          PUSHL   #1                                              ;
                                                01  DD 006C3          PUSHL   #1                                              ;
```

```
                                00000000G  8F   DD 006C5          PUSHL   #VERIFY$_OPENINDEX                                    ;
                    00000000G   00         04   FB 006CB          CALLS   #4, LIB$SIGNAL                                       ;
                                           01   DD 006D2 23$:     PUSHL   #1                                                   ; 0805
                          0000V CF         01   FB 006D4          CALLS   #1, READ_HOMEBLOCK                                   ;
                       4F 00000000'  EF    04   E1 006D9          BBC     #4, QUAL, 24$                                        ; 0810
                       00C00000'  EF       01   90 006E1          MOVB    #1, USAGE_BUFFER                                     ; 0813
          00000000'    EF 00000000'  EF    34   28 006E8          MOVC3   #52, BUFFER+456, USAGE_BUFFER+1                      ; 0817
                                00000000'  EF   9F 006F4          PUSHAB  USAGE_BUFFER+53                                      ; 0818
                    00000000G   00         01   FB 006FA          CALLS   #1, S^S$GETTIM                                       ;
                    00000000'   EF         3D   B0 00701          MOVW    #61, USAGE_RAB+34                                    ; 0819
                                00000000'  EF   9F 00708          PUSHAB  USAGE_RAB                                            ; 0820
                    00000000G   00         01   FB 0070E          CALLS   #1, S^S$PUT                                          ;
                                18         50   E8 00715          BLBS    R0, 24$                                              ;
                                7E 00000000'  EF   7D 00718          MOVQ    USAGE_RAB+8, -(SP)                                   ; 0825
                                00000000'  EF   9F 0071F          PUSHAB  USAGE_FAB                                            ; 0822
                                00000000G  8F   DD 00725          PUSHL   #VERIFY$_FACILITY+4308                               ; 0823
                          0000V CF         04   FB 0072B          CALLS   #4, FILE_ERROR                                       ;
                                56 00000000'  EF   9E 00730 24$:     MOVAB   PER_VOLUME_BEG, R6                                   ; 0833
                                5B 00000000'  EF   9E 00737          MOVAB   PER_VOLUME_END-4, R11                                ;
                       57 00000000'  EF         02   78 0073E          ASHL    #2, VOLUME_COUNT, R7                                 ; 0835
                                                39   11 00746          BRB     27$                                                  ;
                                                56   DD 00748 25$:     PUSHL   A                                                    ;
                                                57   D0 0074A          MOVL    R7, 8(SP)                                            ;
                                08   AE            AE   9F 0074E          PUSHAB  8(SP)                                                ;
                    00000000G   00         02   FB 00751          CALLS   #2, LIB$GET_VM                                       ;
                                59         50   D0 00758          MOVL    R0, STATUS                                           ;
                                11         59   E8 0075B          BLBS    STATUS, 26$                                          ; 0836
                                           59   DD 0075E          PUSHL   STATUS                                               ;
                                           7E   D4 00760          CLRL    -(SP)                                                ;
                                00000000G  8F   DD 00762          PUSHL   #VERIFY$_ALLOCMEM                                    ;
                    00000000G   00         03   FB 00768          CALLS   #3, LIB$SIGNAL                                       ;
                57  00000000'   EF         02   78 0076F 26$:     ASHL    #2, VOLUME_COUNT, R7                                 ; 0837
          57    00             6E         00   2C 00777          MOVC5   #0, (SP), #0, R7, @0(A)                              ;
                                      00   B6      0077C                                                                        ;
                                56         04   C0 0077E          ADDL2   #4, A                                                ; 0833
                                5B         56   D1 00781 27$:     CMPL    A, R11                                               ;
                                           C2   1B 00784          BLEQU   25$                                                  ;
                                           58   DD 00786          PUSHL   ACCTL_0                                              ; 0843
                                           01   DD 00788          PUSHL   #1                                                   ;
                          0000V CF         02   FB 0078A          CALLS   #2, INIT_VOL_DATA                                    ;
                    00000000'   FF         58   D0 0078F          MOVL    ACCTL_0, @ACCTL                                      ; 0844
                                57 00000000'  EF   D0 00796          MOVL    VOLUME_COUNT, R7                                     ; 0851
                                56         01   D0 0079D          MOVL    #1, RVN                                              ;
                                                01   31 007A0          BRW     33$                                                  ;
0040  8F          00             6E         00   2C 007A3 28$:     MOVC5   #0, (SP), #0, #64, FIB                               ; 0853
                                00000000'  EF   007AA                                                                           ;
                                50 00000000'FF46   DE 007AF          MOVAL   @ACCTL[RVN], R0                                      ; 0854
                          FC    A0 00200100  8F   D0 007B7          MOVL    #2097408, -4(R0)                                     ;
                    00000000'   EF 00200100  8F   D0 007BF          MOVL    #2097408, FIB                                        ;
                    00000000'   EF 00010001  8F   D0 007CA          MOVL    #65537, FIB+4                                        ; 0855
                    00000000'   EF           56   B0 007D5          MOVW    RVN, FIB+8                                           ; 0857
                                           7E   D4 007DC          CLRL    -(SP)                                                ; 0863
                                F6E6       CF   9F 007DE          PUSHAB  HDR_ATR_DESC                                         ;
                                           7E   7C 007E2          CLRQ    -(SP)                                                ;
                                           7E   D4 007E4          CLRL    -(SP)                                                ;
                                F6EA       CF   9F 007E6          PUSHAB  FIB_DESC                                             ;
                                           7E   7C 007EA          CLRQ    -(SP)                                                ;
```

```
                                00000000'  EF   9F 007EC                PUSHAB  IOSB                                    ;
                             7E       72  8F   9A 007F2                MOVZBL  #114, -(SP)                             ;
                                00000000'  EF   DD 007F6                PUSHL   CHANNEL                                 ;
                                           7E   D4 007FC                CLRL    -(SP)                                   ;
                   00000000G   00          0C   FB 007FE                CALLS   #12, SYS$QIOW                           ;
                               59          50   D0 00805                MOVL    R0, STATUS                              ;
                               07          59   E9 00808                BLBC    STATUS, 29$                             ; 0864
                               59 00000000'   EF   3C 0080B                MOVZWL  IOSB, STATUS                            ;
                   0000025C    8F          59   D1 00812 29$:           CMPL    STATUS, #604                            ; 0865
                                           72   12 00819                BNEQ    30$                                     ;
    0040    8F          00                 6E   00   2C 0081B           MOVC5   #0, (SP), #0, #64, FIB                  ; 0868
                                00000000'  EF        00822                                                            ;
                                           5A   01   D0 00827           MOVL    #1, NO_WRITE                            ; 0869
                               50 00000000'FF46   DE 0082A              MOVAL   @ACCTL[RVN], R0                         ; 0870
                            FC     A0 00200000   8F   D0 00832           MOVL    #2097152, -4(R0)                        ;
                   00000000'   EF   00200000   8F   D0 0083A             MOVL    #2097152, FIB                           ;
                   00000000'   EF   00010001   8F   D0 00845             MOVL    #65537, FIB+4                           ; 0871
                   00000000'   EF              56   B0 00850             MOVW    RVN, FIB+8                              ; 0873
                                               7E   D4 00857             CLRL    -(SP)                                   ; 0879
                                     F66B      CF   9F 00859             PUSHAB  HDR_ATR_DESC                            ;
                                               7E   7C 0085D             CLRQ    -(SP)                                   ;
                                               7E   D4 0085F             CLRL    -(SP)                                   ;
                                     F66F      CF   9F 00861             PUSHAB  FIB_DESC                                ;
                                               7E   7C 00865             CLRQ    -(SP)                                   ;
                                00000000'      EF   9F 00867             PUSHAB  IOSB                                    ;
                             7E       72      8F   9A 0086D             MOVZBL  #114, -(SP)                             ;
                                00000000'      EF   DD 00871             PUSHL   CHANNEL                                 ;
                                               7E   D4 00877             CLRL    -(SP)                                   ;
                   00000000G   00              0C   FB 00879             CALLS   #12, SYS$QIOW                           ;
                               59              50   D0 00880             MOVL    R0, STATUS                              ;
                               0A              59   E9 00883             BLBC    STATUS, 31$                             ; 0880
                               59 00000000'    EF   3C 00886             MOVZWL  IOSB, STATUS                            ;
                               13              59   E8 0088D 30$:         BLBS    STATUS, 32$                             ; 0882
                                         0240  8F   BB 00890 31$:         PUSHR   #^M<R6,R9>                              ;
                                               01   DD 00894             PUSHL   #1                                      ;
                                00000000G      8F   DD 00896             PUSHL   #VERIFY$_OPENINDEX                      ;
                   00000000G   00              04   FB 0089C             CALLS   #4, LIB$SIGNAL                          ;
                                               56   DD 008A3 32$:         PUSHL   RVN                                     ; 0883
                        0000V  CF              01   FB 008A5             CALLS   #1, READ_HOMEBLOCK                      ;
                               50 00000000'FF46   DE 008AA              MOVAL   @ACCTL[RVN], R0                         ; 0884
                                        FC     A0   DD 008B2             PUSHL   -4(R0)                                  ;
                                               56   DD 008B5             PUSHL   RVN                                     ;
                        0000V  CF              02   FB 008B7             CALLS   #2, INIT_VOL_DATA                       ;
        FEE1            56                     01   F1 008BC 33$:         ACBL    R7, #1, RVN, 28$                        ; 0851
                                               15   E9 008C2             BLBC    NO_WRITE, 34$                           ; 0890
                   0D 00000000'                EF   03   E5 008C5        BBCC    #3, QUAL, 34$                           ; 0892
                                00000000G      8F   DD 008CD             PUSHL   #VERIFY$_NOREPAIR                       ; 0894
                   00000000G   00              01   FB 008D3             CALLS   #1, LIB$SIGNAL                          ;
                                00000000'      EF   D4 008DA 34$:         CLRL    QT                                      ; 0899
                   00000000'   EF       0100  8F   3C 008E0             MOVZWL  #256, QT+4                              ; 0900
                               02 00000000'    EF   D1 008E9             CMPL    STRUCTURE_LEVEL, #2                     ; 0903
                                               03   13 008F0             BEQL    35$                                     ;
                                           01F8   31 008F2              BRW     52$                                     ;
    0040    8F          00                 6E   00   2C 008F5 35$:         MOVC5   #0, (SP), #0, #64, FIB                  ; 0910
                                00000000'  EF        008FC                                                            ;
                   00000000'   EF 00000000'   FF   D0 00901             MOVL    @ACCTL, FIB                             ; 0911
                   00000000'   EF 00040004    8F   D0 0090C             MOVL    #262148, FIB+10                         ; 0912
```

```
                 00000000'  EF                 01  B0 00917          MOVW    #1, FIB+14                      ; 0914
                                                7E  D4 0091E          CLRL    -(SP)                           ; 0921
                                      F590      CF  9F 00920          PUSHAB  FAT_ATR_DESC                    ;
                                                7E  7C 00924          CLRQ    -(SP)                           ;
                                      F5C2      CF  9F 00926          PUSHAB  QFI_DESC                        ;
                                      F5A6      CF  9F 0092A          PUSHAB  FIB_DESC                        ;
                                                7E  7C 0092E          CLRQ    -(SP)                           ;
                                00000000'       EF  9F 00930          PUSHAB  IOSB                            ;
                                   7E     72    8F  9A 00936          MOVZBL  #114, -(SP)                     ;
                                00000000'       EF  DD 0093A          PUSHL   CHANNEL                         ;
                                                7E  D4 00940          CLRL    -(SP)                           ;
                 00000000G      00              0C  FB 00942          CALLS   #12, SYS$QIOW                   ;
                                59              50  D0 00949          MOVL    R0, STATUS                      ;
                                4E              59  E9 0094C          BLBC    STATUS, 38$                     ; 0922
                                59 00000000'    EF  3C 0094F          MOVZWL  IOSB, STATUS                    ;
                                44              59  E9 00956          BLBC    STATUS, 38$                     ; 0924
                                   00000000'    EF  95 00959          TSTB    UCHAR                           ; 0927
                                                1A  18 0095F          BGEQ    36$                             ;
                                01 00000000'    EF  91 00961          CMPB    RECATTR, #1                     ; 0928
                                                11  12 00968          BNEQ    36$                             ;
                                   00000000'    EF  95 0096A          TSTB    RECATTR+1                       ; 0929
                                                09  12 00970          BNEQ    36$                             ;
                                20 00000000'    EF  B1 00972          CMPW    RECATTR+2, #32                  ; 0930
                                                1F  13 00979          BEQL    37$                             ;
                                                7E  7C 0097B 36$:     CLRQ    -(SP)                           ; 0935
                                                7E  7C 0097D          CLRQ    -(SP)                           ;
                                                7E  7C 0097F          CLRQ    -(SP)                           ;
                                                7E  7C 00981          CLRQ    -(SP)                           ;
                                7E              34  7D 00983          MOVQ    #52, -(SP)                      ;
                                   00000000'    EF  DD 00986          PUSHL   CHANNEL                         ;
                                                7E  D4 0098C          CLRL    -(SP)                           ;
                 00000000G      00              0C  FB 0098E          CALLS   #12, SYS$QIOW                   ;
                                59      03BC    8F  3C 00995          MOVZWL  #956, STATUS                    ; 0936
                                14              59  E8 0099A 37$:     BLBS    STATUS, 39$                     ; 0939
                                                59  DD 0099D 38$:     PUSHL   STATUS                          ; 0941
                                                7E  D4 0099F          CLRL    -(SP)                           ;
                                   00000000G    8F  DD 009A1          PUSHL   #VERIFY$_OPENQUOTA              ;
                 00000000G      00              03  FB 009A7          CALLS   #3, LIB$SIGNAL                  ;
                                             013C   31 009AE          BRW     52$                             ;
                 00000000'      EF              01  D0 009B1 39$:     MOVL    #1, QUOTA_ACTIVE                ; 0947
              50 00000000'      EF              10  9C 009B8          ROTL    #16, RECATTR+8, R0              ; 0952
                                5B         FF   A0  9E 009C0          MOVAB   -1(R0), R11                     ;
                                58              3F  CE 009C4          MNEGL   #63, VBN                        ;
                                             00FF   31 009C7          BRW     51$                             ;
              50 00000000'      EF              10  9C 009CA 40$:     ROTL    #16, RECATTR+8, R0              ; 0962
                                50              58  C2 009D2          SUBL2   VBN, R0                         ;
                 00000040       8F              50  D1 009D5          CMPL    R0, #64                         ; 0960
                                                04  1B 009DC          BLEQU   41$                             ;
                                50         40   8F  9A 009DE          MOVZBL  #64, R0                         ;
                                57              50  D0 009E2 41$:     MOVL    R0, THIS_BLOCKS                 ;
                                                7E  7C 009E5          CLRQ    -(SP)                           ; 0975
                                                7E  D4 009E7          CLRL    -(SP)                           ;
                                                58  DD 009E9          PUSHL   VBN                             ;
              5A                57              09  78 009EB          ASHL    #9, THIS_BLOCKS, R10            ;
                                                5A  DD 009EF          PUSHL   R10                             ;
                                   00000000'    EF  9F 009F1          PUSHAB  BUFFER                          ;
                                                7E  7C 009F7          CLRQ    -(SP)                           ;
```

```
                                00000000'   EF   9F  009F9          PUSHAB  IOSB
                                            31   DD  009FF          PUSHL   #49
                                00000000'   EF   DD  00A01          PUSHL   CHANNEL
                                            7E   D4  00A07          CLRL    -(SP)
                    00000000G   00           0C   FB  00A09          CALLS   #12, SYS$QIOW
                                59           50   D0  00A10          MOVL    R0, STATUS
                                0A           59   E9  00A13          BLBC    STATUS, 42$                     0976
                                59 00000000' EF   3C  00A16          MOVZWL  IOSB, STATUS
                                6A           59   E8  00A1D          BLBS    STATUS, 46$                     0977
                                56           01   CE  00A20 42$:     MNEGL   #1, XVBN                        0980
                                             61   11  00A23          BRB     45$
                                             7E   7C  00A25 43$:     CLRQ    -(SP)                           0988
                                             7E   D4  00A27          CLRL    -(SP)
                                          6648   9F  00A29          PUSHAB  (XVBN)[VBN]
                                7E      0200   8F   3C  00A2C          MOVZWL  #512, -(SP)
                    52          56           09   78  00A31          ASHL    #9, XVBN, R2
                                00000000'EF42   9F  00A35          PUSHAB  BUFFER[R2]
                                             7E   7C  00A3C          CLRQ    -(SP)
                                00000000'   EF   9F  00A3E          PUSHAB  IOSB
                                            31   DD  00A44          PUSHL   #49
                                00000000'   EF   DD  00A46          PUSHL   CHANNEL
                                            7E   D4  00A4C          CLRL    -(SP)
                    00000000G   00           0C   FB  00A4E          CALLS   #12, SYS$QIOW
                                59           50   D0  00A55          MOVL    R0, STATUS
                                0A           59   E9  00A58          BLBC    STATUS, 44$                     0989
                                59 00000000' EF   3C  00A5B          MOVZWL  IOSB, STATUS
                                21           59   E8  00A62          BLBS    STATUS, 45$                     0990
                                             59   DD  00A65 44$:     PUSHL   STATUS                          0993
                                          6648   9F  00A67          PUSHAB  (XVBN)[VBN]
                                             01   DD  00A6A          PUSHL   #1
                                00000000G   8F   DD  00A6C          PUSHL   #VERIFY$_READQUOTA
                    00000000G   00           04   FB  00A72          CALLS   #4, LIB$SIGNAL
    0200    8F      00          6E           00   2C  00A79          MOVC5   #0, (SP), #0, #512, BUFFER[R2]  0994
                                00000000'EF42        00A80
                    98          56           57   F2  00A86 45$:     AOBLSS  THIS_BLOCKS, XVBN, 43$          0980
                                52 00000000' EF   9E  00A8A 46$:     MOVAB   BUFFER, R2                      1002
                                53 00000000'EF4A  9E  00A91          MOVAB   BUFFER-32[R10], R3
                                             29   11  00A99          BRB     50$
                                23           62   E9  00A9B 47$:     BLBC    (ENTRY), 49$                    1007
                                50        04 A2   D0  00A9E          MOVL    4(ENTRY), R0                    1010
                                             0A   12  00AA2          BNEQ    48$
                        00000000' EF      0C A2   7D  00AA4          MOVQ    12(ENTRY), DEFAULT_QUOTA        1013
                                             13   11  00AAC          BRB     49$                             1010
                        00000000' EF              50   D0  00AAE 48$:     MOVL    R0, LAST_UIC                1018
                                             7E   D4  00AB5          CLRL    -(SP)                           1019
                                  08 A2      DD   00AB7          PUSHL   8(ENTRY)
                                             50   DD  00ABA          PUSHL   R0
                                0000V        CF   03  00ABC          CALLS   #3, COUNT_QUOTA
                                52           20   C0  00AC1 49$:     ADDL2   #32, ENTRY                      1002
                                53           52   D1  00AC4 50$:     CMPL    ENTRY, R3
                                             D2   1B  00AC7          BLEQU   47$
        FEF7    58 00000040    8F           5B   F1  00AC9 51$:     ACBL    R11, #64, VBN, 40$              0952
                                             7E   7C  00AD3          CLRQ    -(SP)                           1030
                                             7E   7C  00AD5          CLRQ    -(SP)
                                             7E   7C  00AD7          CLRQ    -(SP)
                                             7E   7C  00AD9          CLRQ    -(SP)
                                7E           34   7D  00ADB          MOVQ    #52, -(SP)
```

```
                                 00000000'  EF   DD 00ADE                PUSHL   CHANNEL                                          ;
                                            7E   D4 00AE4                CLRL    -(SP)                                            ;
                          00000000G  00     0C   FB 00AE6                CALLS   #12, SYS$QIOW                                    ;
                                 00000000'  EF   9F 00AED 52$:           PUSHAB  CURRENT_TIME                                     ; 1037
                          00000000G  00     01   FB 00AF3                CALLS   #1, SYS$GETTIM                                   ;
                                            7E   D4 00AFA                CLRL    -(SP)                                            ; 1038
                                 00000000'  EF   9F 00AFC                PUSHAB  CURRENT_TIME                                     ;
                                      F4DA  CF   9F 00B02                PUSHAB  P.ABF                                            ;
                                            7E   D4 00B06                CLRL    -(SP)                                            ;
                          00000000G  00     04   FB 00B08                CALLS   #4, SYS$ASCTIM                                   ;
                          00000000'  EF 00000000'EF  B0 00B0F            MOVW    BUFFER, CURRENT_TIME_1                           ; 1039
00000000'  EF       18                00 00000000'EF  F0 00B1A            INSV    BUFFER+3, #0, #24, CURRENT_TIME_1+2              ; 1040
                          00000000'  EF 00000000'EF  B0 00B27            MOVW    BUFFER+9, CURRENT_TIME_1+5                       ; 1041
                          00000000'  EF 00000000'EF  B0 00B32            MOVW    BUFFER+12, CURRENT_TIME_1+7                      ; 1042
                          00000000'  EF 00000000'EF  B0 00B3D            MOVW    BUFFER+15, CURRENT_TIME_1+9                      ; 1043
                          00000000'  EF 00000000'EF  B0 00B48            MOVW    BUFFER+18, CURRENT_TIME_1+11                     ; 1044
                                    20 00000000'EF  91 00B53            CMPB    CURRENT_TIME_1, #32                              ; 1045
                                               07   12 00B5A            BNEQ    53$                                              ;
                          00000000'  EF        30   90 00B5C            MOVB    #48, CURRENT_TIME_1                              ;
                               0000V CF        00   FB 00B63 53$:       CALLS   #0, SCAN_INDEX                                   ; 1050
                                  04 AE 00000000'  EF  D0 00B68         MOVL    VOLUME_COUNT, 4(SP)                              ; 1055
                                            56   D4 00B70                CLRL    RVN                                              ;
                                          03F5   31 00B72                BRW     91$                                              ;
    0040  8F        00                    6E        00   2C 00B75 54$:   MOVC5   #0, (SP), #0, #64, FIB                          ; 1063
                                 00000000'  EF        00B7C                                                                      ;
                                    50 00000000'FF46   DE 00B81         MOVAL   @ACCTL[RVN], R0                                  ; 1064
                          00000000'  EF       FC   A0  D0 00B89         MOVL    -4(R0), FIB                                      ;
                          00000000'  EF 00020002   8F  D0 00B91         MOVL    #131074, FIB+4                                   ; 1065
                          00000000'  EF            56  B0 00B9C         MOVW    RVN, FIB+8                                       ; 1067
                                            7E   7C 00BA3                CLRQ    -(SP)                                            ; 1072
                                            7E   7C 00BA5                CLRQ    -(SP)                                            ;
                                            7E   D4 00BA7                CLRL    -(SP)                                            ;
                                          F327   CF  9F 00BA9            PUSHAB  FIB_DESC                                         ;
                                            7E   7C 00BAD                CLRQ    -(SP)                                            ;
                                 00000000'  EF   9F 00BAF                PUSHAB  IOSB                                             ;
                                       7E   72   8F  9A 00BB5            MOVZBL  #114, -(SP)                                      ;
                                 00000000'  EF   DD 00BB9                PUSHL   CHANNEL                                          ;
                                            7E   D4 00BBF                CLRL    -(SP)                                            ;
                          00000000G  00     0C   FB 00BC1                CALLS   #12, SYS$QIOW                                    ;
                                    59              50  D0 00BC8         MOVL    R0, STATUS                                       ;
                                    0A              59  E9 00BCB         BLBC    STATUS, 55$                                      ; 1073
                                    59 00000000'    EF  3C 00BCE         MOVZWL  IOSB, STATUS                                     ;
                                    13              59  E8 00BD5         BLBS    STATUS, 56$                                      ; 1074
                                         0240       8F  BB 00BD8 55$:    PUSHR   #^M<R6,R9>                                       ; 1076
                                            01      DD 00BDC             PUSHL   #1                                               ;
                                 00000000G  8F      DD 00BDE             PUSHL   #VERIFY$_OPENBITMAP                              ;
                          00000000G  00     04   FB 00BE4                CALLS   #4, LIB$SIGNAL                                   ;
                                       08   AE  9F 00BEB 56$:           PUSHAB  SMAP                                             ; 1081
                                    50 00000000'FF46   DE 00BEE         MOVAL   @SMAP_SIZE[RVN], R0                              ;
04  AE        FC                    A0              09  78 00BF6         ASHL    #9, -4(R0), 4(SP)                                ;
                                       04   AE  9F 00BFC                PUSHAB  4(SP)                                            ;
                          00000000G  00     02   FB 00BFF                CALLS   #2, LIB$GET_VM                                   ;
                                    59              50  D0 00C06         MOVL    R0, STATUS                                       ;
                                    11              59  E8 00C09         BLBS    STATUS, 57$                                      ; 1082
                                                    59  DD 00C0C         PUSHL   STATUS                                           ;
                                                    7E  D4 00C0E         CLRL    -(SP)                                            ;
                                 00000000G  8F      DD 00C10             PUSHL   #VERIFY$_ALLOCMEM                                ;
```

```
                    00000000G  00                03  FB 00C16          CALLS   #3, LIB$SIGNAL
                               50 00000000'FF46   DE 00C1D 57$:       MOVAL   @SMAP_SIZE[RVN], R0          1087
                            57       FC  A0       01  C3 00C25          SUBL3   #1, -4(R0), R7
                               52       81  8F     8F  98 00C2A          CVTBL   #-127, VBN                   1109
                                          0089     31  00C2E          BRW     64$
                               50 00000000'FF46   DE 00C31 58$:       MOVAL   @SMAP_SIZE[RVN], R0          1097
                            50       FC  A0       52  C3 00C39          SUBL3   VBN, -4(R0), R0
                        0000007F        8F       50  D1 00C3E          CMPL    R0, #127                     1095
                                                  04  1B 00C45          BLEQU   59$
                               50       7F  8F     9A  00C47          MOVZBL  #127, R0
                               55           50     D0  00C4B 59$:       MOVL    R0, THIS_BLOCKS
                                            7E     7C  00C4E          CLRQ    -(SP)                        1109
                                            7E     D4  00C50          CLRL    -(SP)
                                       02   A2     9F  00C52          PUSHAB  2(VBN)
                            7E             55     09  78 00C55          ASHL    #9, THIS_BLOCKS, -(SP)
                            50             52     09  78 00C59          ASHL    #9, VBN, R0
                            54             50   1C    AE  C1 00C5D          ADDL3   SMAP, R0, R4
                                            54     DD  00C62          PUSHL   R4
                                            7E     7C  00C64          CLRQ    -(SP)
                                  00000000'        EF  9F 00C66          PUSHAB  IOSB
                                            31     DD  00C6C          PUSHL   #49
                                  00000000'        EF  DD 00C6E          PUSHL   CHANNEL
                                            7E     D4  00C74          CLRL    -(SP)
                    00000000G  00                0C  FB 00C76          CALLS   #12, SYS$QIOW
                               59                50  D0 00C7D          MOVL    R0, STATUS
                               0A                59  E9 00C80          BLBC    STATUS, 60$                  1110
                               59 00000000'      EF  3C 00C83          MOVZWL  IOSB, STATUS
                               5D                59  E8 00C8A          BLBS    STATUS, 64$                  1111
                               53                01  CE 00C8D 60$:       MNEGL   #1, XVBN                     1114
                                                 54  11 00C90          BRB     63$
                                            7E   7C  00C92 61$:       CLRQ    -(SP)                        1122
                                            7E   D4  00C94          CLRL    -(SP)
                                       02 A342   9F  00C96          PUSHAB  2(XVBN)[VBN]
                                  7E       0200   8F  CC 00C9A          MOVZWL  #512, -(SP)
                            50             53     09  78 00C9F          ASHL    #9, XVBN, R0
                                         6044     9F  00CA3          PUSHAB  (R0)[R4]
                                            7E     7C  00CA6          CLRQ    -(SP)
                                  00000000'        EF  9F 00CA8          PUSHAB  IOSB
                                            31     DD  00CAE          PUSHL   #49
                                  00000000'        EF  DD 00CB0          PUSHL   CHANNEL
                                            7E     D4  00CB6          CLRL    -(SP)
                    00000000G  00                0C  FB 00CB8          CALLS   #12, SYS$QIOW
                               59                50  D0 00CBF          MOVL    R0, STATUS
                               0A                59  E9 00CC2          BLBC    STATUS, 62$                  1123
                               59 00000000'      EF  3C 00CC5          MOVZWL  IOSB, STATUS
                               17                59  E8 00CCC          BLBS    STATUS, 63$                  1124
                                       0240      8F  BB 00CCF 62$:       PUSHR   #^M<R6,R9>                   1130
                                       02 A342   9F  00CD3          PUSHAB  2(XVBN)[VBN]                 1129
                                            02   DD  00CD7          PUSHL   #2                           1126
                                  00000000G        8F  DD 00CD9          PUSHL   #VERIFY$_READSBMAP
                    00000000G  00                05  FB 00CDF          CALLS   #5, LIB$SIGNAL
                            A8 53                55  F2 00CE6 63$:       AOBLSS  THIS_BLOCKS, XVBN, 61$       1114
                FF3D        52 0000007F          8F  F1 00CEA 64$:       ACBL    R7, #127, VBN, 58$           1087
                               50 00000000'FF46   DE 00CF4          MOVAL   @SMAP_SIZE[RVN], R0          1139
                            5B       FC  A0       0C  78 00CFC          ASHL    #12, -4(R0), R11
                               52                01  CE 00D01          MNEGL   #1, J
                                                 0080  31 00D04          BRW     71$
```

```
  5A      08   BE          01            52  EF 00D07 65$:   EXTZV   J, #1, @SMAP, R10                     ; 1141
                           51 00000000'FF46  DE 0000D         MOVAL   @VSMAP[RVN], R1                       ;
  55      FC   B1          01            52  EF 00D15         EXTZV   J, #1, @-4(R1), R5                    ;
                           55            5A  D1 0001B         CMPL    R10, R5                               ;
                                         67  13 0001E         BEQL    71$                                   ;
                           50 00000000'FF46  DE 00D20         MOVAL   @SMAP_SIZE[RVN], R0                   ; 1147
               50      FC  A0            0C  78 00D28         ASHL    #12, -4(R0), R0                       ;
                           58      FF    A2  9E 00D2D         MOVAB   -1(R2), K                             ;
                                         19  11 00D31         BRB     67$                                   ;
  57      08   BE          01            58  EF 00D33 66$:    EXTZV   K, #1, @SMAP, R7                      ; 1150
  54      FC   B1          01            58  EF 00D39         EXTZV   K, #1, @-4(R1), R4                    ;
                           54            57  D1 00D3F         CMPL    R7, R4                                ;
                                         0C  13 00D42         BEQL    68$                                   ;
               55                        54  D1 00D44         CMPL    R4, R5                                ; 1151
                                         07  12 00D47         BNEQ    68$                                   ;
                           53            58  D0 00D49         MOVL    K, KK                                 ; 1154
               E3          58            50  F2 00D4C 67$:    AOBLSS  R0, K, 66$                            ; 1147
                                         56  DD 00D50 68$:    PUSHL   RVN                                   ; 1164
                           51      01    A3  9E 00D52         MOVAB   1(R3), R1                             ; 1163
                           50 00000000'FF46  DE 00D56         MOVAL   @CLUSTER_FACTOR[RVN], R0              ;
                           51      FC    A0  C4 00D5E         MULL2   -4(R0), R1                            ;
                                   FF    A1  9F 00D62         PUSHAB  -1(R1)                                ;
               7E          52      FC    A0  C5 00D65         MULL3   -4(R0), J, -(SP)                      ; 1162
                                         03  DD 00D6A         PUSHL   #3                                    ; 1157
               08                        5A  E9 00D6C         BLBC    R10, 69$                              ; 1158
                              00000000G  8F  DD 00D6F         PUSHL   #VERIFY$_ALLOCSET                     ;
                                         06  11 00D75         BRB     70$                                   ;
                              00000000G  8F  DD 00D77 69$:    PUSHL   #VERIFY$_ALLOCCLR                     ;
                    00000000G  00        05  FB 00D7D 70$:    CALLS   #5, LIB$SIGNAL                        ;
                           52            53  D0 00D84         MOVL    KK, J                                 ; 1166
               02          52            5B  F2 00D87 71$:    AOBLSS  R11, J, 72$                           ; 1139
                                         03  11 00D8B         BRB     73$                                   ;
                                       FF77  31 00D8D 72$:    BRW     65$                                   ;
                           50 00000000'FF46  DE 00D90 73$:    MOVAL   @SMAP_SIZE[RVN], R0                   ; 1174
               5B      FC  A0            0C  78 00D98         ASHL    #12, -4(R0), R11                      ;
                           52            01  CE 00D9D         MNEGL   #1, J                                 ;
                                         7D  11 00DA0         BRB     78$                                   ;
                           54 00000000'FF46  DE 00DA2 74$:    MOVAL   @NSMAP[RVN], R4                       ; 1176
  50      FC   B4          01            52  EF 00DAA         EXTZV   J, #1, @-4(R4), R0                    ;
                           51 00000000'FF46  DE 00DB0         MOVAL   @VSMAP[RVN], R1                       ;
  57      FC   B1          01            52  EF 00DB8         EXTZV   J, #1, @-4(R1), R7                    ;
                           57            50  D1 00DBE         CMPL    R0, R7                                ;
                                         5C  13 00DC1         BEQL    78$                                   ;
                           50 00000000'FF46  DE 00DC3         MOVAL   @SMAP_SIZE[RVN], R0                   ; 1182
               50      FC  A0            0C  78 00DCB         ASHL    #12, -4(R0), R0                       ;
                           5A      FF    A2  9E 00DD0         MOVAB   -1(R2), K                             ;
                                         19  11 00DD4         BRB     76$                                   ;
  58      FC   B4          01            5A  EF 00DD6 75$:    EXTZV   K, #1, @-4(R4), R8                    ; 1185
  55      FC   B1          01            5A  EF 00DDC         EXTZV   K, #1, @-4(R1), R5                    ;
                           55            58  D1 00DE2         CMPL    R8, R5                                ;
                                         0C  13 00DE5         BEQL    77$                                   ;
                           57            55  D1 00DE7         CMPL    R5, R7                                ; 1186
                                         07  12 00DEA         BNEQ    77$                                   ;
                           53            5A  D0 00DEC         MOVL    K, KK                                 ; 1189
               E3          5A            50  F2 00DEF 76$:    AOBLSS  R0, K, 75$                            ; 1182
                                         56  DD 00DF3 77$:    PUSHL   RVN                                   ; 1197
                           51      01    A3  9E 00DF5         MOVAB   1(R3), R1                             ; 1196
```

```
                        50 00000000'FF46   DE 00DF9           MOVAL   @CLUSTER_FACTOR[RVN], R0
                        51       FC   A0   C4 00E01           MULL2   -4(R0), R1
                        FF       A1   9F   00E05              PUSHAB  -1(R1)
                7E              52   FC   A0   C5 00E08       MULL3   -4(R0), J, -(SP)                1195
                                     03   DD 00E0D            PUSHL   #3                              1192
                            00000000G    8F   DD 00E0F       PUSHL   #VERIFY$_ALLOCEXT
                    00000000G   00       05   FB 00E15       CALLS   #5, LIB$SIGNAL
                        52               53   D0 00E1C       MOVL    KK, J                           1199
                02              52           5B   F2 00E1F 78$:  AOBLSS R11, J, 79$                  1174
                                         03   11 00E23       BRB     80$
                                     FF7A     31 00E25 79$:  BRW     74$
                                   08   AE   9F 00E28 80$:  PUSHAB  SMAP                             1206
                        50 00000000'FF46   DE 00E2B           MOVAL   @SMAP_SIZE[RVN], R0
            04  AE          FC   A0       09   78 00E33       ASHL    #9, -4(R0), 4(SP)
                                   04   AE   9F 00E39         PUSHAB  4(SP)
                    00000000G   00       02   FB 00E3C       CALLS   #2, LIB$FREE_VM
                        59               50   D0 00E43       MOVL    R0, STATUS
                        11               59   E8 00E46       BLBS    STATUS, 81$                     1207
                                         59   DD 00E49       PUSHL   STATUS
                                         7E   D4 00E4B       CLRL    -(SP)
                                00000000G    8F   DD 00E4D   PUSHL   #VERIFY$_FREEMEM
                    00000000G   00           03   FB 00E53   CALLS   #3, LIB$SIGNAL
                                         7E   D4 00E5A 81$:  CLRL    -(SP)                           1212
                        0000V   CF           01   FB 00E5C   CALLS   #1, DO_REPAIR
                        03               50   E8 00E61       BLBS    R0, 82$
                                     00E9     31 00E64       BRW     90$
                        50 00000000'FF46   DE 00E67 82$:  MOVAL   @SMAP_SIZE[RVN], R0                1215
            57          FC   A0           01   C3 00E6F       SUBL3   #1, -4(R0), R7
                        52       81   8F   98 00E74           CVTBL   #-127, VBN
                                     00CB     31 00E78       BRW     89$
                        50 00000000'FF46   DE 00E7B 83$:  MOVAL   @SMAP_SIZE[RVN], R0                1225
        50          FC   A0               52   C3 00E83       SUBL3   VBN, -4(R0), R0
            0000007F    8F               50   D1 00E88       CMPL    R0, #127                        1223
                                         04   1B 00E8F       BLEQU   84$
                        50       7F      8F   9A 00E91       MOVZBL  #127, R0
                        55               50   D0 00E95 84$:  MOVL    R0, THIS_BLOCKS
                                         7E   7C 00E98       CLRQ    -(SP)                           1237
                                         7E   D4 00E9A       CLRL    -(SP)
                                   02    A2   9F 00E9C       PUSHAB  2(VBN)
            7E                  55           09   78 00E9F   ASHL    #9, THIS_BLOCKS, -(SP)
                        50 00000000'FF46   DE 00EA3           MOVAL   @NSMAP[RVN], R0
            54          52               09   78 00EAB       ASHL    #9, VBN, R4
                              FC B044    9F 00EAF             PUSHAB  @-4(R0)[R4]
                                         7E   7C 00EB3       CLRQ    -(SP)
                            00000000'    EF   9F 00EB5       PUSHAB  IOSB
                                         30   DD 00EBB       PUSHL   #48
                            00000000'    EF   DD 00EBD       PUSHL   CHANNEL
                                         7E   D4 00EC3       CLRL    -(SP)
                    00000000G   00           0C   FB 00EC5   CALLS   #12, SYS$QIOW
                        59               50   D0 00ECC       MOVL    R0, STATUS
                        0A               59   E9 00ECF       BLBC    STATUS, 85$                     1238
                        59 00000000'     EF   3C 00ED2       MOVZWL  IOSB, STATUS
                        6A               59   E8 00ED9       BLBS    STATUS, 89$                     1239
                        53               01   CE 00EDC 85$:  MNEGL   #1, XVBN                        1242
                                         61   11 00EDF       BRB     88$
                                         7E   7C 00EE1 86$:  CLRQ    -(SP)                           1250
                                         7E   D4 00EE3       CLRL    -(SP)
```

```
                                    02 A342   9F 00EE5           PUSHAB  2(XVBN)[VBN]                                    ;
                              7E      0200   8F  3C 00EE9       MOVZWL  #512, -(SP)                                     ;
                              50 00000000'FF46  DE 00EEE       MOVAL   @NSMAP[RVN], R0                                 ;
                        50    54        FC   A0  C1 00EF6      ADDL3   -4(R0), R4, R0                                  ;
                        51    53             09  78 00EFB      ASHL    #9, XVBN, R1                                    ;
                                          6140  9F 00EFF       PUSHAB  (R1)[R0]                                        ;
                                             7E  7C 00F02       CLRQ    -(SP)                                           ;
                                 00000000'   EF  9F 00F04       PUSHAB  IOSB                                            ;
                                             30  DD 00F0A       PUSHL   #48                                             ;
                                 00000000'   EF  DD 00F0C       PUSHL   CHANNEL                                         ;
                                             7E  D4 00F12       CLRL    -(SP)                                           ;
                     00000000G   00              0C  FB 00F14   CALLS   #12, SYS$QIOW                                   ;
                              59             50  D0 00F1B       MOVL    R0, STATUS                                      ;
                              0A             59  E9 00F1E       BLBC    STATUS, 87$                                     ; 1251
                              59 00000000'   EF  3C 00F21       MOVZWL  IOSB, STATUS                                    ;
                              17             59  E8 00F28       BLBS    STATUS, 88$                                     ; 1252
                                    0240     8F  BB 00F2B 87$:  PUSHR   #^M<R6,R9>                                     ; 1258
                                       02 A342  9F 00F2F       PUSHAB  2(XVBN)[VBN]                                    ; 1257
                                             02  DD 00F33       PUSHL   #2                                              ; 1254
                                 00000000G   8F  DD 00F35       PUSHL   #VERIFY$_WRITESBMAP                             ;
                     00000000G   00              05  FB 00F3B   CALLS   #5, LIB$SIGNAL                                 ;
                        9B       53             55  F2 00F42 88$:  AOBLSS  THIS_BLOCKS, XVBN, 86$                      ; 1242
              FF2B      52 0000007F   8F         57  F1 00F46 89$:  ACBL    R7, #127, VBN, 83$                          ; 1215
                                             7E  7C 00F50 90$:  CLRQ    -(SP)                                           ; 1270
                                             7E  7C 00F52       CLRQ    -(SP)                                           ;
                                             7E  7C 00F54       CLRQ    -(SP)                                           ;
                                             7E  7C 00F56       CLRQ    -(SP)                                           ;
                              7E             34  7D 00F58       MOVQ    #52, -(SP)                                      ;
                                 00000000'   EF  DD 00F5B       PUSHL   CHANNEL                                         ;
                                             7E  D4 00F61       CLRL    -(SP)                                           ;
                     00000000G   00              0C  FB 00F63   CALLS   #12, SYS$QIOW                                   ;
              FC04      56             01        04   AE  F1 00F6A 91$:  ACBL    4(SP), #1, RVN, 54$                     ; 1055
                              0C 00000000'   EF  E9 00F71       BLBC    DUAL_ALLOC_FOUND, 92$                           ; 1276
                     00000000'   EF              01  D0 00F78   MOVL    #1, DUAL_ALLOC_PASS                             ; 1279
                          0000V  CF              00  FB 00F7F   CALLS   #0, SCAN_INDEX                                  ; 1280
                              55 00000000'   EF  D0 00F84 92$:  MOVL    VOLUME_COUNT, R5                                ; 1286
                                             53  D4 00F8B       CLRL    RVN                                             ;
                                          ^^98   31 00F8D       BRW     96$                                             ;
                              50 00000000   r43  DE 00F90 93$:  MOVAL   @MAXFILIDX[RVN], R0                             ; 1288
                        54       FC   A0             01  C1 00F98  ADDL3   #1, -4(R0), R4                              ;
                                             52  D4 00F9D       CLRL    NUM                                             ;
                                          0083   31 00F9F       BRW     95$                                             ;
                              50 00000000'FF43  DE 00FA2 94$:  MOVAL   @IMAP[RVN], R0                                  ; 1291
                              51        FF   A2  9E 00FAA       MOVAB   -1(R2), R1                                      ;
                        72       FC   B0             51  E1 00FAE  BBC     R1, @-4(R0), 95$                          ;
                              50 00000000'FF43  DE 00FB3       MOVAL   @LOSTMAP[RVN], R0                               ; 1292
                        65       FC   B0             51  E0 00FBB  BBS     R1, @-4(R0), 95$                          ;
                              50 00000000'FF43  DE 00FC0       MOVAL   @EXTMAP[RVN], R0                                ; 1293
                        58       FC   B0             51  E0 00FC8  BBS     R1, @-4(R0), 95$                          ;
                              64        AE             52  B0 00FCD  MOVW    NUM, FILE_ID                              ; 1299
                  50    52             08             10  EF 00FD1  EXTZV   #16, #8, NUM, R0                          ; 1300
                              69        AE             50  90 00FD6  MOVB    R0, FILE_ID+5                             ;
                              50 00000000'FF43  DE 00FDA       MOVAL   @SEQMAP[RVN], R0                                ; 1301
                              66        AE       FC B041  B0 00FE2  MOVW    @-4(R0)[R1], FILE_ID+2                    ;
                              68        AE             53  90 00FE8  MOVB    RVN, FILE_ID+4                            ; 1302
                                             7E  D4 00FEC       CLRL    -(SP)                                           ; 1303
                                       68    AE  9F 00FEE       PUSHAB  FILE_ID                                         ;
```

```
                                00000000G   8F   DD 00FF1          PUSHL   #VERIFY$_LOSTEXTHDR                      ;
                          0000V CF          03   FB 00FF7          CALLS   #3, HEADER_ERROR                        ;
                                            7E   D4 00FFC          CLRL    -(SP)                                   ; 1304
                          0000V CF          01   FB 00FFE          CALLS   #1, DO_REPAIR                           ;
                                1F          50   E9 01003          BLBC    R0, 95$                                 ;
                                00000000'   EF   9F 01006          PUSHAB  BUFFER_2                                ; 1306
                                      68    AE   9F 0100C          PUSHAB  FILE_ID                                 ;
                          0000V CF          02   FB 0100F          CALLS   #2, READ_HEADER                         ;
                                0E          50   E9 01014          BLBC    R0, 95$                                 ;
                                00000000'   EF   9F 01017          PUSHAB  BUFFER_2                                ; 1308
                                      68    AE   9F 0101D          PUSHAB  FILE_ID                                 ;
                          0000V CF          02   FB 01020          CALLS   #2, DELETE_HEADER                       ;
        FF77        52                    01   F1 01025 95$:     ACBL    R4, #1, NUM, 94$                        ; 1288
        FF5F        53                    01   F1 0102B 96$:     ACBL    R5, #1, RVN, 93$                        ; 1286
                                            7E   D4 01031          CLRL    -(SP)                                   ; 1316
                          0000V CF          01   FB 01033          CALLS   #1, DO_REPAIR                           ;
                                03          50   E8 01038          BLBS    R0, 97$                                 ;
                                      01A7  31   01038          BRW     110$                                    ;
                                58 00000000'EF   D0 0103E 97$:     MOVL    VOLUME_COUNT, R8                        ; 1318
                                            56   D4 01045          CLRL    RVN                                     ;
                                      0195  31   01047          BRW     109$                                    ;
    0040    8F          00          6E      00   2C 0104A 98$:     MOVC5   #0, (SP), #0, #64, FIB                  ; 1323
                                00000000'   EF      01051                                                        ;
                   00000000'    EF 00200100 8F   D0 01056          MOVL    #2097408, FIB                           ; 1324
                   00000000'    EF 00010001 8F   D0 01061          MOVL    #65537, FIB+4                           ; 1325
                   00000000'    EF          56   B0 0106C          MOVW    RVN, FIB+8                              ; 1327
                                            7E   7C 01073          CLRQ    -(SP)                                   ; 1332
                                            7E   7C 01075          CLRQ    -(SP)                                   ;
                                            7E   D4 01077          CLRL    -(SP)                                   ;
                                      EE57  CF   9F 01079          PUSHAB  FIB_DESC                                ;
                                            7E   7C 0107D          CLRQ    -(SP)                                   ;
                                00000000'   EF   9F 0107F          PUSHAB  IOSB                                    ;
                                   7E   72  8F   9A 01085          MOVZBL  #114, -(SP)                             ;
                                00000000'   EF   DD 01089          PUSHL   CHANNEL                                 ;
                                            7E   D4 0108F          CLRL    -(SP)                                   ;
                   00000000G    00          0C   FB 01091          CALLS   #12, SYS$QIOW                           ;
                                59          50   D0 01098          MOVL    R0, STATUS                              ;
                                0A          59   E9 0109B          BLBC    STATUS, 99$                             ; 1333
                                59 00000000'EF   3C 0109E          MOVZWL  IOSB, STATUS                            ;
                                13          59   E8 010A5          BLBS    STATUS, 100$                            ; 1334
                                      0240  8F   BB 010A8 99$:      PUSHR   #^M<R6,R9>                              ;
                                            01   DD 010AC          PUSHL   #1                                      ;
                                00000000G   8F   DD 010AE          PUSHL   #VERIFY$_OPENINDEX                      ;
                   00000000G    00          04   FB 010B4          CALLS   #4, LIB$SIGNAL                          ;
                                50 00000000'FF46  DE 010BB 100$:    MOVAL   @IMAP_SIZE[RVN], R0                     ; 1339
    57      FC      A0                    01   C3 010C3          SUBL3   #1, -4(R0), R7                          ;
                                52      81  8F   98 010C8          CVTBL   #-127, VBN                              ;
                                      00EC  31   010CC 101$:      BRW     108$                                    ;
                                50 00000000'FF46  DE 010CF 102$:    MOVAL   @IMAP_SIZE[RVN], R0                     ; 1349
    50      FC      A0                    52   C3 010D7          SUBL3   VBN, -4(R0), R0                         ;
            0000007F    8F                50   D1 010DC          CMPL    R0, #127                                ; 1347
                                            04   1B 010E3          BLEQU   103$                                    ;
                                50      7F  8F   9A 010E5          MOVZBL  #127, R0                                ;
                                55          50   D0 010E9 103$:    MOVL    R0, THIS_BLOCKS                         ;
                                            7E   7C 010EC          CLRQ    -(SP)                                   ; 1361
                                            7E   D4 010EE          CLRL    -(SP)                                   ;
                                50 00000000'FF46  DE 010F0         MOVAL   @BITMAP_OFFSET[RVN], R0                 ;
```

```
                                FC B042   9F 010F8          PUSHAB  @-4(R0)[VBN]                                    ;
                             7E      55      09 78 010FC          ASHL    #9, THIS_BLOCKS, -(SP)                          ;
                             50 00000000'FF46  DE 01100          MOVAL   @IMAP[RVN], R0                                  ;
                             54      52      09 78 01108          ASHL    #9, VBN, R4                                     ;
                                FC B044   9F 0110C          PUSHAB  @-4(R0)[R4]                                     ;
                                      7E  7C 01110          CLRQ    -(SP)                                           ;
                                00000000'  EF 9F 01112          PUSHAB  IOSB                                            ;
                                      30  DD 01118          PUSHL   #48                                             ;
                                00000000'  EF DD 0111A          PUSHL   CHANNEL                                         ;
                                      7E  D4 01120          CLRL    -(SP)                                           ;
                    00000000G   00          0C FB 01122          CALLS   #12, SYS$QIOW                                   ;
                             59      50      D0 01129          MOVL    R0, STATUS                                      ;
                             0A      59      E9 0112C          BLBC    STATUS, 104$                                    ; 1362
                             59 00000000'  EF 3C 0112F          MOVZWL  IOSB, STATUS                                    ;
                             93      59      E8 01136          BLBS    STATUS, 101$                                    ; 1363
                             53      01      CE 01139 104$:     MNEGL   #1, XVBN                                        ; 1366
                                      79  11 0113C          BRB     107$                                            ;
                                      7E  7C 0113E 105$:     CLRQ    -(SP)                                           ; 1374
                                      7E  D4 01140          CLRL    -(SP)                                           ;
                             50 00000000'FF46  DE 01142          MOVAL   @BITMAP_OFFSET[RVN], R0                         ;
                          50      52      FC  A0 C1 01146          ADDL3   -4(R0), VBN, R0                                 ;
                                    6340  9F 0114F          PUSHAB  (XVBN)[R0]                                      ;
                                7E      0200  8F 3C 01152          MOVZWL  #512, -(SP)                                     ;
                             50 00000000'FF46  DE 01157          MOVAL   @IMAP[RVN], R0                                  ;
                          50      54      FC  A0 C1 0115F          ADDL3   -4(R0), R4, R0                                  ;
                          51      53      09 78 01164          ASHL    #9, XVBN, R1                                    ;
                                    6140  9F 01168          PUSHAB  (R1)[R0]                                        ;
                                      7E  7C 0116B          CLRQ    -(SP)                                           ;
                                00000000'  EF 9F 0116D          PUSHAB  IOSB                                            ;
                                      30  DD 01173          PUSHL   #48                                             ;
                                00000000'  EF DD 01175          PUSHL   CHANNEL                                         ;
                                      7E  D4 0117B          CLRL    -(SP)                                           ;
                    00000000G   00          0C FB 0117D          CALLS   #12, SYS$QIOW                                   ;
                             59      50      D0 01184          MOVL    R0, STATUS                                      ;
                             0A      59      E9 01187          BLBC    STATUS, 106$                                    ; 1375
                             59 00000000'  EF 3C 0118A          MOVZWL  IOSB, STATUS                                    ;
                             23      59      E8 01191          BLBS    STATUS, 107$                                    ; 1376
                                0240  8F BB 01194 106$:     PUSHR   #^M<R6,R9>                                      ; 1382
                             50 00000000'FF46  DE 01198          MOVAL   @BITMAP_OFFSET[RVN], R0                         ; 1381
                          50      52      FC  A0 C1 011A0          ADDL3   -4(R0), VBN, R0                                 ;
                                    6340  9F 011A5          PUSHAB  (XVBN)[R0]                                      ;
                                      02  DD 011A8          PUSHL   #2                                              ; 1378
                                00000000G  8F DD 011AA          PUSHL   #VERIFY$_WRITEIBMAP                             ;
                    00000000G   00          05 FB 011B0          CALLS   #5, LIB$SIGNAL                                  ;
                          83      53      55 F2 011B7 107$:     AOBLSS  THIS_BLOCKS, XVBN, 105$                         ; 1366
FF0A                      52 0000007F   8F  57 F1 011BB 108$:     ACBL    R7, #127, VBN, 102$                             ; 1339
                                      7E  7C 011C5          CLRQ    -(SP)                                           ; 1393
                                      7E  7C 011C7          CLRQ    -(SP)                                           ;
                                      7E  7C 011C9          CLRQ    -(SP)                                           ;
                                      7E  7C 011CB          CLRQ    -(SP)                                           ;
                                   7E     34 7D 011CD          MOVQ    #52, -(SP)                                      ;
                                00000000'  EF DD 011D0          PUSHL   CHANNEL                                         ;
                                      7E  D4 011D6          CLRL    -(SP)                                           ;
                    00000000G   00          0C FB 011D8          CALLS   #12, SYS$QIOW                                   ;
FE65                      56    01          58 F1 011DF 109$:     ACBL    R8, #1, RVN, 98$                                ; 1318
                             53 00000000'  EF D0 011E5 110$:     MOVL    VOLUME_COUNT, R3                                ; 1399
                                      52  D4 011EC          CLRL    RVN                                             ;
```

```
                                                07  11 011EE             BRB     112$                                                    ;
                                                52  DD 011F0 111$:       PUSHL   RVN                                                     ; 1401
                                      0000V     CF     01  FB 011F2             CALLS   #1, DIR_SCAN                                    ;
                                         F5     52     53  F3 011F7 112$:       AOBLEQ  R3, RVN, 111$                                   ; 1399
                                                0D 00000000'   FF  E9 011FB     BLBC    DIRECTORY_ERROR, 113$                           ; 1407
                                                   00000000G   8F  DD 01202             PUSHL   #VERIFY$_LOSTSCAN                       ;
                                            00000000G   00     01  FB 01208             CALLS   #1, LIB$SIGNAL                          ;
                                                  04    AE 00000000'   EF  D0 0120F 113$:       MOVL    VOLUME_COUNT, 4(SP)             ; 1410
                                                                57  D4 01217             CLRL    RVN                                     ;
                                                          018A  31 01219             BRW     127$                                    ;
                                                50 00000000'FF47  DE 0121C 114$:       MOVAL   @MAXFILIDX[RVN], R0                     ; 1412
                                         6E        FC     A0     01  C1 01224             ADDL3   #1, -4(R0), (SP)                        ;
                                                                56  D4 01229             CLRL    NUM                                     ;
                                                          0172  31 0122B 115$:       BRW     126$                                    ;
                                                50 00000000'FF47  DE 0122E 116$:       MOVAL   @LOSTMAP[RVN], R0                       ; 1414
                                                58        FF    A6  9E 01236             MOVAB   -1(R6), R8                              ;
                                         EC        FC     B0     58  E1 0123A             BBC     R8, @-4(R0), 115$                       ;
                                                  64    AE     56  B0 0123F             MOVW    NUM, FILE_ID                            ; 1423
                                   50            56     08     10  EF 01243             EXTZV   #16, #8, NUM, R0                        ; 1424
                                                  69    AE     50  90 01248             MOVB    R0, FILE_ID+5                           ;
                                                50 00000000'FF47  DE 0124C             MOVAL   @SEQMAP[RVN], R0                        ; 1425
                                                  66    AE FC B048  B0 01254             MOVW    @-4(R0)[R8], FILE_ID+2                  ;
                                                  68    AE     57  90 0125A             MOVB    RVN, FILE_ID+4                          ; 1426
                                      03 00000000'    EF     04  E0 0125E             BBS     #4, QUAL, 118$                          ; 1431
                                                          0101  31 01266 117$:       BRW     123$                                    ;
                                                00000000'   EF  9F 01269 118$:       PUSHAB  BUFFER_2                                ;
                                                      68    AE  9F 0126F             PUSHAB  FILE_ID                                 ;
                                               0000V  CF        02  FB 01272             CALLS   #2, READ_HEADER                         ;
                                                      EC        50  E9 01277             BLBC    R0, 117$                                ;
                                                5A 00000000'    EF  9A 0127A             MOVZBL  BUFFER_2, R10                           ; 1440
                                                5B 00000000'EF4A  3E 01281             MOVAW   BUFFER_2[R10], IDENT_AREA               ;
                                                02 00000000'    EF  D1 01289             CMPL    STRUCTURE_LEVEL, #2                     ; 1441
                                                               3C  12 01290             BNEQ    121$                                    ;
0057    8F                 20                   6B                  14  2C 01292             MOVC5   #20, (IDENT_AREA), #32, #87, FILENAME   ; 1445
                                                         0C    AE        01299                                                             ;
                                                50 00000000'    EF  9A 0129B             MOVZBL  BUFFER_2+1, R0                          ; 1449
                                                50              5A  C2 012A2             SUBL2   R10, R0                                 ;
                                                50              02  C4 012A5             MULL2   #2, R0                                  ;
                                            00000078    8F          50  D1 012AB             CMPL    R0, #120                        ; 1450
                                                                    08  1F 012AF             BLSSU   119$                            ;
                                   20    AE       36    AB     0042    8F  28 012B1             MOVC3   #66, 54(IDENT_AREA), FILENAME+20        ; 1455
                                   0C    AE     0057    8F             20  3A 012B9 119$:       LOCC    #32, #87, FILENAME              ; 1458
                                                                       02  12 012C0             BNEQ    120$                            ;
                                                                       51  D4 012C2             CLRL    R1                              ;
                                                50        0C    AE     9E 012C4 120$:       MOVAB   FILENAME, R0                    ; 1459
                                         50             51           50  C3 012C8             SUBL3   R0, R1, LENGTH                  ;
                                                                       0D  11 012CC             BRB     122$                            ; 1441
                                                          0C    AE     9F 012CE 121$:       PUSHAB  FILENAME                        ; 1463
                                                          FA    AB     9F 012D1             PUSHAB  -6(IDENT_AREA)                  ; 1464
                                            00000000G   EF             02  FB 012D4             CALLS   #2, MAKE_STRING                 ;
                                            00000000'   EF             02  90 012DB 122$:       MOVB    #2, USAGE_BUFFER                ; 1469
                                                51 00000000'FF47  DE 012E2             MOVAL   @OWNER[RVN], R1                         ; 1470
                                            00000000'   EF        FC B148  D0 012EA             MOVL    @-4(R1)[R8], USAGE_BUFFER+1     ;
                                                51 00000000'FF47  DE 012F3             MOVAL   @ALLOCATION[RVN], R1                    ; 1471
                                            00000000'   EF        FC B148  D0 012FB             MOVL    @-4(R1)[R8], USAGE_BUFFER+5     ;
                                                51 00000000'FF47  DE 01304             MOVAL   @USAGE[RVN], R1                         ; 1472
                                            00000000'   EF        FC B148  D0 0130C             MOVL    @-4(R1)[R8], USAGE_BUFFER+9     ;
```

```
                        00000000'  EF                  02  B0  01315          MOVW    #2, USAGE_BUFFER+13                     ; 1473
                                                   0C   AE  9F  0131C          PUSHAB  FILENAME                                ; 1480
                                                        50  DD  0131F          PUSHL   LENGTH
                                                  ECD3  CF  9F  01321          PUSHAB  P.ABI
                                           00000000'    EF  9F  01325          PUSHAB  USAGE_BUFFER+15
                                                ECC1    CF  9F  0132B          PUSHAB  P.ABG
                                  00000000G     00        05  FB  0132F          CALLS   #5, SYS$FAO
                00000000'  EF     00000000'     EF        11  A1  01336          ADDW3   #17, USAGE_BUFFER+15, USAGE_RAB+34       ; 1481
                                           00000000'    EF  9F  01342          PUSHAB  USAGE_RAB                               ; 1482
                                  00000000G     00        01  FB  01348          CALLS   #1, SYS$PUT
                                                18        50  E8  0134F          BLBS    R0, 123$
                                                7E  00000000'  EF  7D  01352    MOVQ    USAGE_RAB+8, -(SP)                      ; 1487
                                                    00000000'  EF  9F  01359    PUSHAB  USAGE_FAB                               ; 1484
                                                    00000000G  8F  DD  0135F    PUSHL   #VERIFY$_FACILITY+4308                  ; 1485
                                         0000V      CF        04  FB  01365          CALLS   #4, FILE_ERROR
                                                2F  00000000'  EF  E8  0136A  123$:  BLBS    DIRECTORY_ERROR, 126$                   ; 1493
                                                7E              D4  01371          CLRL    -(SP)                                   ; 1496
                                                    68          AE  9F  01373          PUSHAB  FILE_ID
                                                    00000000G   8F  DD  01376          PUSHL   #VERIFY$_LOSTHEADER
                                         0000V      CF        03  FB  0137C          CALLS   #3, HEADER_ERROR
                                                               03  DD  01381          PUSHL   #3                                      ; 1497
                                         0000V      CF        01  FB  01383          CALLS   #1, DO_REPAIR
                                                59             50  D0  01388          MOVL    R0, STATUS
                                                12             59  E9  0138B          BLBC    STATUS, 126$
                                                           64  AE  9F  0138E          PUSHAB  FILE_ID                                 ; 1499
                                  04            59             01  E1  01391          BBC     #1, STATUS, 124$                        ; 1500
                                                               03  DD  01395          PUSHL   #3
                                                               02  11  01397          BRB     125$
                                                               7E  D4  01399  124$:    CLRL    -(SP)
                                         0000V      CF        02  FB  0139B  125$:    CALLS   #2, ENTER_WORK
                        FE88     56                01        6E  F1  013A0  126$:    ACBL    (SP), #1, NUM, 116$                     ; 1412
                        FE6F     57                01  04    AE  F1  013A6  127$:    ACBL    4(SP), #1, RVN, 114$                    ; 1410
                                                    03  00000000'  EF  E8  013AD          BLBS    QUOTA_ACTIVE, 128$               ; 1512
                                                            016F  31  013B4          BRW     145$
                                                               7E  D4  013B7  128$:    CLRL    -(SP)                                   ; 1521
                                         0000V      CF        01  FB  013B9          CALLS   #1, DO_REPAIR
                                                7B             50  E9  013BE          BLBC    R0, 13T$
                0040     8F     00             6E        00  2C  013C1          MOVC5   #0, (SP), #0, #64, FIB                  ; 1528
                                                    00000000'  EF      013C8
                                  00000000'     EF  00040004  8F  D0  013CD          MOVL    #262148, FIB+10                         ; 1529
                                  00000000'     EF            01  B0  013D8          MOVW    #1, FIB+14                              ; 1531
                                  00000000'     EF            09  B0  013DF          MOVW    #9, FIB+22                              ; 1532
                                                               7E  7C  013E6          CLRQ    -(SP)                                   ; 1538
                                                               7E  7C  013E8          CLRQ    -(SP)
                                                          EAFE  CF  9F  013EA          PUSHAB  QFI_DESC
                                                          EAE2  CF  9F  013EE          PUSHAB  FIB_DESC
                                                               7E  7C  013F2          CLRQ    -(SP)
                                                    00000000'  EF  9F  013F4          PUSHAB  IOSB
                                                               38  DD  013FA          PUSHL   #56
                                                    00000000'  EF  DD  013FC          PUSHL   CHANNEL
                                                               7E  D4  01402          CLRL    -(SP)
                                  00000000G     00            0C  FB  01404          CALLS   #12, SYS$QIOW
                                                59             50  D0  0140B          MOVL    R0, STATUS
                                                07             59  E9  0140E          BLBC    STATUS, 129$                            ; 1539
                                                59  00000000'  EF  3C  01411          MOVZWL  IOSB, STATUS
                                  000003CC      8F            59  D1  01418  129$:    CMPL    STATUS, #972                            ; 1540
                                                               1B  13  0141F          BEQL    131$
```

```
                        11              59  E8 01421          BLBS    STATUS, 130$                        ; 1543
                                        59  DD 01424          PUSHL   STATUS
                                        7E  D4 01426          CLRL    -(SP)
                            00000000G   8F  DD 01428          PUSHL   #VERIFY$_ENAQUOTA
                  00000000G 00          03  FB 0142E          CALLS   #3, LIB$SIGNAL
                  00000000' EF                  01  D0 01435 130$:    MOVL    #1, QUOTA_DISABLE           ; 1544
                            57                  01  D0 0143C 131$:    MOVL    #1, M                       ; 1549
                                00000000'   EF  D5 0143F          TSTL    LAST_UIC                    ; 1550
                                            02  12 01445          BNEQ    132$
                                            57  D4 01447          CLRL    M
                            58 00000000'    EF  D0 01449 132$:    MOVL    QT, T                       ; 1551
                                            03  12 01450 133$:    BNEQ    134$                        ; 1554
                                        00D1    31 01452          BRW     145$
                            56      08      A8  9E 01455 134$:    MOVAB   8(R8), E                    ; 1556
                            5A              01  CE 01459          MNEGL   #1, J                       ; 1557
                                        00B7    31 0145C          BRW     142$
                      08    A6      04      A6  D1 0145F 135$:    CMPL    4(E), 8(E)                  ; 1559
                                            03  12 01464          BNEQ    137$
                                        009F    31 01466 136$:    BRW     140$
                                            66  DD 01469 137$:    PUSHL   (E)                         ; 1570
                            7E      04      A6  7D 0146B          MOVQ    4(E), -(SP)                 ; 1568
                                            03  DD 0146F          PUSHL   #3                          ; 1565
                                00000000G   8F  DD 01471          PUSHL   #VERIFY$_INCQUOTA
                  00000000G 00              05  FB 01477          CALLS   #5, LIB$SIGNAL
                      0000V CF              00  FB 0147E          CALLS   #0, DO_REPAIR               ; 1571
                            E0              50  E9 01483          BLBC    R0, 136$
                            73              57  E9 01486          BLBC    M, 139$                     ; 1573
    0040    8F      00                      6E  00 2C 01489          MOVC5   #0, (SP), #0, #64, FIB      ; 1579
                                00000000'   EF      01490
                  00000000'     EF              0D  B0 01495          MOVW    #13, FIB+22                 ; 1580
                  00000000'     EF              04  D0 0149C          MOVL    #4, FIB+24                  ; 1581
                  00000000'     EF              66  D0 014A3          MOVL    (E), DQF+4                  ; 1582
                  00000000'     EF      08      A6  D0 014AA          MOVL    8(E), DQF+8                 ; 1583
                                                7E  7C 014B2          CLRQ    -(SP)                       ; 1589
                                                7E  7C 014B4          CLRQ    -(SP)
                                            EA22    CF  9F 014B6          PUSHAB  DQF_DESC
                                            EA16    CF  9F 014BA          PUSHAB  FIB_DESC
                                                7E  7C 014BE          CLRQ    -(SP)
                                    00000000'   EF  9F 014C0          PUSHAB  IOSB
                                                38  DD 014C6          PUSHL   #56
                                    00000000'   EF  DD 014C8          PUSHL   CHANNEL
                                                7E  D4 014CE          CLRL    -(SP)
                  00000000G 00                  0C  FB 014D0          CALLS   #12, SYS$QIOW
                            59                  50  D0 014D7          MOVL    R0, STATUS
                            0A                  59  E9 014DA          BLBC    STATUS, 138$                ; 1590
                            59 00000000'    EF  3C 014DD          MOVZWL  IOSB, STATUS
                            21                  59  E8 014E4          BLBS    STATUS, 140$                ; 1591
                                                59  DD 014E7 138$:    PUSHL   STATUS                      ; 1597
                                                66  DD 014E9          PUSHL   (E)                         ; 1596
                                                01  DD 014EB          PUSHL   #1                          ; 1593
                                    00000000G   8F  DD 014ED          PUSHL   #VERIFY$_MODQUOTA
                  00000000G 00                  04  FB 014F3          CALLS   #4, LIB$SIGNAL
                                                0C  11 014FA          BRB     140$                        ; 1573
                                            08      A6  DD 014FC 139$:    PUSHL   8(E)                    ; 1604
                                                66  DD 014FF          PUSHL   (E)
                                                02  DD 01501          PUSHL   #2
                      0000V CF                  03  FB 01503          CALLS   #3, ENTER_WORK
```

```
                        00000000'  EF                    66  D1 01508 140$:   CMPL    (E), LAST_UIC                          ; 1609
                                                          02  12 0150F         BNEQ    141$                                   ;
                                                          57  D4 01511         CLRL    M                                      ;
                                   56                     0C  C0 01513 141$:   ADDL2   #12, E                                 ; 1610
                              02   5A            04       A8  F2 01516 142$:   AOBLSS  4(T), J, 143$                          ; 1557
                                                          03  11 0151B         BRB     144$                                   ;
                                                    FF3F  31 0151D 143$:      BRW     135$                                   ;
                                   58                     68  D0 01520 144$:   MOVL    (T), T                                 ; 1612
                                                    FF2A  31 01523         BRW     133$                                   ; 1554
                           5E 00000000'  EF                03  E1 01526 145$:   BBC     #3, QUAL, 148$                         ; 1619
0040  8F                   00            6E                00  2C 0152E         MOVC5   #0, (SP), #0, #64, FIB                 ; 1622
                                                00000000'  EF    01535
                        00000000'  EF                    08  B0 0153A         MOVW    #8, FIB+22                             ; 1623
                                                          7E  7C 01541         CLRQ    -(SP)                                  ; 1628
                                                          7E  7C 01543         CLRQ    -(SP)
                                                          7E  D4 01545         CLRL    -(SP)                                  ;
                                                E989      CF  9F 01547         PUSHAB  FIB_DESC                               ;
                                                          7E  7C 0154B         CLRQ    -(SP)                                  ;
                                             00000000'  EF  9F 0154D         PUSHAB  IOSB                                   ;
                                                          38  DD 01553         PUSHL   #56                                    ;
                                             00000000'  EF  DD 01555         PUSHL   CHANNEL                                ;
                                                          7E  D4 0155B         CLRL    -(SP)                                  ;
                        00000000G  00                     0C  FB 0155D         CALLS   #12, SYS$QIOW                          ;
                                   59                     50  D0 01564         MOVL    R0, STATUS                             ;
                                   0A                     59  E9 01567         BLBC    STATUS, 146$                           ; 1629
                                   59  00000000'          EF  3C 0156A         MOVZWL  IOSB, STATUS                           ;
                                   11                     59  E8 01571         BLBS    STATUS, 147$                           ; 1630
                                                          59  DD 01574 146$:   PUSHL   STATUS                                 ;
                                                          7E  D4 01576         CLRL    -(SP)                                  ;
                                             00000000G  8F  DD 01578         PUSHL   #VERIFY$_UNLKVOL                       ;
                        00000000G  00                     03  FB 0157E         CALLS   #3, LIB$SIGNAL                         ;
                        00000000'  EF                     08  8A 01585 147$:   BICB2   #8, QUAL                               ; 1631
                             0000V CF                     00  FB 0158C 148$:   CALLS   #0, PROCESS_WORK                       ; 1637
                                   5D  00000000'          EF  E9 01591         BLBC    QUOTA_DISABLE, 151$                    ; 1642
0040  8F                   00            6E                00  2C 01598         MOVC5   #0, (SP), #0, #64, FIB                 ; 1645
                                             00000000'  EF    0159F
                        00000000'  EF                    0A  B0 015A4         MOVW    #10, FIB+22                            ; 1646
                                                          7E  7C 015AB         CLRQ    -(SP)                                  ; 1651
                                                          7E  7C 015AD         CLRQ    -(SP)
                                                          7E  D4 015AF         CLRL    -(SP)                                  ;
                                                E91F      CF  9F 015B1         PUSHAB  FIB_DESC                               ;
                                                          7E  7C 015B5         CLRQ    -(SP)                                  ;
                                             00000000'  EF  9F 015B7         PUSHAB  IOSB                                   ;
                                                          38  DD 015BD         PUSHL   #56                                    ;
                                             00000000'  EF  DD 015BF         PUSHL   CHANNEL                                ;
                                                          7E  D4 015C5         CLRL    -(SP)                                  ;
                        00000000G  00                     0C  FB 015C7         CALLS   #12, SYS$QIOW                          ;
                                   59                     50  D0 015CE         MOVL    R0, STATUS                             ;
                                   0A                     59  E9 015D1         BLBC    STATUS, 149$                           ; 1652
                                   59  00000000'          EF  3C 015D4         MOVZWL  IOSB, STATUS                           ;
                                   11                     59  E8 015DB         BLBS    STATUS, 150$                           ; 1653
                                                          59  DD 015DE 149$:   PUSHL   STATUS                                 ; 1655
                                                          7E  D4 015E0         CLRL    -(SP)                                  ;
                                             00000000G  8F  DD 015E2         PUSHL   #VERIFY$_DSAQUOTA                      ;
                        00000000G  00                     03  FB 015E8         CALLS   #3, LIB$SIGNAL                         ;
                                             00000000'  EF  D4 015EF 150$:   CLRL    QUOTA_DISABLE                          ; 1656
                                                          7E  D4 015F5 151$:   CLRL    -(SP)                                  ; 1663
```

```
                    0000V  CF                01  FB 015F7              CALLS   #1, ACCESS_INDEX_2
                           00000000'         EF  DD 015FC              PUSHL   CHANNEL                                  ;  1664
                00000000G  00                01  FB 01602              CALLS   #1, SYS$DASSGN
                           00000000'         EF  DD 01609              PUSHL   CHANNEL_2                                ;  1665
                00000000G  00                01  FB 0160F              CALLS   #1, SYS$DASSGN
                      28 00000000'  EF       01  E1 01616              BBC     #1, QUAL, 152$                           ;  1670
                           00000000'         EF  9F 0161E              PUSHAB  LIST_FAB                                 ;  1672
                00000000G  00                01  FB 01624              CALLS   #1, SYS$CLOSE
                           18                50  E8 0162B              BLBS    R0, 152$
                           7E 00000000'      EF  7D 0162E              MOVQ    LIST_FAB+8, -(SP)                        ;  1677
                           00000000'         EF  9F 01635              PUSHAB  LIST_FAB                                 ;  1674
                           00000000*         8F  DD 0163B              PUSHL   #<<<VERIFY$_FACILITY@16>+4184>+4>         ;  1675
                    0000V  CF                04  FB 01641              CALLS   #4, FILE_ERROR
                      28 00000000'  EF       04  E1 01646  152$:       BBC     #4, QUAL, 153$                           ;  1682
                           00000000'         EF  9F 0164E              PUSHAB  USAGE_FAB                                ;  1684
                00000000G  00                01  FB 01654              CALLS   #1, SYS$CLOSE
                           18                50  E8 0165B              BLBS    R0, 153$
                           7E 00000000'      EF  7D 0165E              MOVQ    USAGE_FAB+8, -(SP)                       ;  1689
                           00000000'         EF  9F 01665              PUSHAB  USAGE_FAB                                ;  1686
                           00000000*         8F  DD 0166B              PUSHL   #<<<VERIFY$_FACILITY@16>+4184>+4>         ;  1687
                    0000V  CF                04  FB 01671              CALLS   #4, FILE_ERROR
                           50                01  D0 01676  153$:       MOVL    #1, R0                                   ;  1695
                                             04  01679                 RET

; Routine Size:  5754 bytes,    Routine Base:  CODE + 015C
```

```
;  1700     1696  1 ROUTINE INIT_VOL_DATA(RVN,ACCTL_VALUE): NOVALUE=
;  1701     1697  1
;  1702     1698  1 !++
;  1703     1699  1 !
;  1704     1700  1 ! FUNCTIONAL DESCRIPTION:
;  1705     1701  1 !       This routine initializes all the per-volume data during the initial
;  1706     1702  1 !       pass over the volumes and does validation of the basic file structure.
;  1707     1703  1 !       It finishes by deaccessing the channel.
;  1708     1704  1 !
;  1709     1705  1 ! INPUT PARAMETERS:
;  1710     1706  1 !       RVN             - Relative volume number.
;  1711     1707  1 !       ACCTL_VALUE     - Value for FIB$L_ACCTL.
;  1712     1708  1 !
;  1713     1709  1 ! IMPLICIT INPUTS:
;  1714     1710  1 !       BUFFER          - Contains the home block.
;  1715     1711  1 !       HDR_BUFFER      - Contains the index file header.
;  1716     1712  1 !
;  1717     1713  1 ! OUTPUT PARAMETERS:
;  1718     1714  1 !       NONE
;  1719     1715  1 !
;  1720     1716  1 ! IMPLICIT OUTPUTS:
;  1721     1717  1 !       Variables between PER_VOLUME_BEG and PER_VOLUME_END initialized.
;  1722     1718  1 !
;  1723     1719  1 ! ROUTINE VALUE:
;  1724     1720  1 !       NONE
;  1725     1721  1 !
;  1726     1722  1 ! SIDE EFFECTS:
;  1727     1723  1 !       Channel deaccessed.
;  1728     1724  1 !
;  1729     1725  1 !--
;  1730     1726  1
;  1731     1727  2 BEGIN
;  1732     1728  2 LOCAL
;  1733     1729  2         INDEX_FILE_ID:  BBLOCK[FID$C_LENGTH],   ! File ID of index file
;  1734     1730  2         BITMAP_FILE_ID: BBLOCK[FID$C_LENGTH],   ! File ID of bitmap file
;  1735     1731  2         CLUSTER,                                ! Cluster factor
;  1736     1732  2         WINDOW:         REF BBLOCK,             ! Pointer to window block
;  1737     1733  2         STATUS;                                 ! Status variable
;  1738     1734  2
;  1739     1735  2 MACRO
;  1740     1736  2 M       CHECK_LBN(LBN,VBN)=
;  1741     1737  2 M           BEGIN
;  1742     1738  2 M           LOCAL
;  1743     1739  2 M               TESTLBN;
;  1744     1740  2 M
;  1745     1741  2 M           IF NOT MAP_VIRTUAL(.WINDOW, (VBN), TESTLBN)
;  1746     1742  2 M               THEN TRUE
;  1747     1743  2 M               ELSE .TESTLBN NEQ (LBN)
;  1748     1744  2 M           END %,
;  1749     1745  2
;  1750     1746  2 M       INITIALIZE(VALUE,LENGTH,DST)=
;  1751     1747  2 M           BEGIN
;  1752     1748  2 M           LOCAL
;  1753     1749  2 M               P;
;  1754     1750  2 M
;  1755     1751  2 M           P = (DST);
;  1756     1752  2 M           DECR N FROM (LENGTH)-1 TO 0 DO
```

```
;  1757        M 1753  2                 BEGIN
;  1758        M 1754  2                 .P = (VALUE);
;  1759        M 1755  2                 P = .P + 4;
;  1760        M 1756  2                 END;
;  1761          1757  2             END %,
;  1762          1758  2
;  1763        M 1759  2         MOVE(LENGTH,SRC,DST)=
;  1764        M 1760  2             BEGIN
;  1765        M 1761  2             LOCAL
;  1766        M 1762  2                 P, Q;
;  1767        M 1763  2
;  1768        M 1764  2             P = (SRC);  Q = (DST);
;  1769        M 1765  2             DECR N FROM (LENGTH)-1 TO 0 DO
;  1770        M 1766  2                 BEGIN
;  1771        M 1767  2                 .Q = .P;
;  1772        M 1768  2                 P = .P + 4;  Q = .Q + 4;
;  1773        M 1769  2                 END;
;  1774          1770  2             END %;
;  1775          1771  2
;  1776          1772  2
;  1777          1773  2 ! Initialize index file ID and bitmap file ID.
;  1778          1774  2 !
;  1779          1775  2 INDEX_FILE_ID[FID$W_NUM] = FID$C_INDEXF;
;  1780          1776  2 INDEX_FILE_ID[FID$W_SEQ] = FID$C_INDEXF;
;  1781          1777  2 INDEX_FILE_ID[FID$W_RVN] = .RVN;
;  1782          1778  2 BITMAP_FILE_ID[FID$W_NUM] = FID$C_BITMAP;
;  1783          1779  2 BITMAP_FILE_ID[FID$W_SEQ] = FID$C_BITMAP;
;  1784          1780  2 BITMAP_FILE_ID[FID$W_RVN] = .RVN;
;  1785          1781  2
;  1786          1782  2
;  1787          1783  2 ! Get device characteristics for this volume.
;  1788          1784  2 !
;  1789          1785  2 CH$FILL(0, DVI_LENGTH, DEVICE_CHAR);
;  1790        P 1786  2 STATUS = $GETDVI(
;  1791        P 1787  2     DEVNAM=DEVICE_DESC,
;  1792        P 1788  2     ITMLST=UPLIT(
;  1793        P 1789  2         WORD(4, DVI$_MAXBLOCK),
;  1794        P 1790  2         LONG(DEVICE_CHAR[DVI_MAXBLOCK], 0),
;  1795        P 1791  2         WORD(4, DVI$_SECTORS),
;  1796        P 1792  2         LONG(DEVICE_CHAR[DVI_SECTORS], 0),
;  1797        P 1793  2         WORD(4, DVI$_TRACKS),
;  1798        P 1794  2         LONG(DEVICE_CHAR[DVI_TRACKS], 0),
;  1799        P 1795  2         WORD(4, DVI$_CYLINDERS),
;  1800        P 1796  2         LONG(DEVICE_CHAR[DVI_CYLINDERS], 0),
;  1801        P 1797  2         WORD(16, DVI$_NEXTDEVNAM),
;  1802        P 1798  2         LONG(DEVICE_NAME, DEVICE_DESC),
;  1803          1799  2         LONG(0)));
;  1804          1800  2 IF NOT .STATUS
;  1805          1801  2 THEN
;  1806          1802  2     SIGNAL(VERIFY$_GETDVI, 1, .RVN, .STATUS);
;  1807          1803  2
;  1808          1804  2
;  1809          1805  2 ! Initialize information from the home block.
;  1810          1806  2 !
;  1811          1807  2 IF .STRUCTURE_LEVEL EQL 2
;  1812          1808  2 THEN
;  1813          1809  3     BEGIN
```

```
;  1814          1810  3          CLUSTER = .BUFFER[HM2$W_CLUSTER];
;  1815          1811  3          IMAP_SIZE[.RVN-1] = .BUFFER[HM2$W_IBMAPSIZE];
;  1816          1812  3          MAXFILIDX[.RVN-1] = .BUFFER[HM2$W_IBMAPSIZE] * 4096 - 1;
;  1817          1813  3          CLUSTER_FACTOR[.RVN-1] = .CLUSTER;
;  1818          1814  3          BITMAP_OFFSET[.RVN-1] = .CLUSTER*4 + 1;
;  1819          1815  3          HEADER_OFFSET[.RVN-1] = .CLUSTER*4 + .BUFFER[HM2$W_IBMAPSIZE];
;  1820          1816  3          END
;  1821          1817  2      ELSE
;  1822          1818  2          BEGIN
;  1823          1819  2          CLUSTER = 1;
;  1824          1820  2          IMAP_SIZE[.RVN-1] = .BUFFER[HM1$W_IBMAPSIZE];
;  1825          1821  2          MAXFILIDX[.RVN-1] = .BUFFER[HM1$W_IBMAPSIZE] * 4096 - 1;
;  1826          1822  2          CLUSTER_FACTOR[.RVN-1] = 1;
;  1827          1823  2          BITMAP_OFFSET[.RVN-1] = 3;
;  1828          1824  2          HEADER_OFFSET[.RVN-1] = 2 + .BUFFER[HM1$W_IBMAPSIZE];
;  1829          1825  2          END;
;  1830          1826  2      SMAP_SIZE[.RVN-1] =
;  1831          1827  2          ((.DEVICE_CHAR[DVI_MAXBLOCK] + .CLUSTER_FACTOR[.RVN-1] - 1) /
;  1832          1828  2          .CLUSTER_FACTOR[.RVN-1] + 4095) / 4096;
;  1833          1829  2
;  1834          1830  2
;  1835          1831  2      ! Get a window block for the index file.
;  1836          1832  2      !
;  1837          1833  2      WINDOW = CREATE_WINDOW(HDR_BUFFER, .RVN);
;  1838          1834  2
;  1839          1835  2
;  1840        U 1836  2      %if false %then ! Following removed until we understand how to recover
;  1841        U 1837  2
;  1842        U 1838  2
;  1843        U 1839  2      ! Validate the primary home block.  Tests that have already been passed in
;  1844        U 1840  2      ! READ_HOMEBLOCK need not be repeated.
;  1845        U 1841  2      !
;  1846        U 1842  2      IF
;  1847        U 1843  2          BEGIN
;  1848        U 1844  2          IF .STRUCTURE_LEVEL EQL 2
;  1849        U 1845  2          THEN
;  1850        U 1846  2              BEGIN
;  1851        U 1847  2              !
;  1852        U 1848  2              ! The following fields have already been verified:
;  1853        U 1849  2              !       HM2$B_STRUCLEV, HM2$W_CLUSTER,
;  1854        U 1850  2              !       HM2$W_CHECKSUM1, HM2$W_CHECKSUM2.
;  1855        U 1851  2              !
;  1856        U 1852  2              ! The following fields are considered entirely free values:
;  1857        U 1853  2              !       HM2$W_DEVTYPE, HM2$W_VOLCHAR, HM2$L_VOLOWNER, HM2$L_SEC_MASK,
;  1858        U 1854  2              !       HM2$W_PROTECT, HM2$W_FILEPROT, HM2$W_RECPROT, HM2$B_WINDOW,
;  1859        U 1855  2              !       HM2$B_LRU_LIM, HM2$W_EXTEND, HM2$L_SERIALNUM.
;  1860        U 1856  2              !
;  1861        U 1857  2              CHECK_LBN(.BUFFER[HM2$L_HOMELBN], .HOMEVBN) OR
;  1862        U 1858  2              CHECK_LBN(.BUFFER[HM2$L_ALHOMELBN], .BUFFER[HM2$W_ALHOMEVBN]) OR
;  1863        U 1859  2              CHECK_LBN(.BUFFER[HM2$L_ALTIDXLBN], .CLUSTER*3+1) OR
;  1864        U 1860  2              .BUFFER[HM2$B_STRUCVER] EQL 0 OR
;  1865        U 1861  2              .BUFFER[HM2$W_HOMEVBN] NEQ .HOMEVBN OR
;  1866        U 1862  2              .BUFFER[HM2$W_ALHOMEVBN] LEQU .CLUSTER*2 OR
;  1867        U 1863  2              .BUFFER[HM2$W_ALHOMEVBN] GTRU .CLUSTER*3 OR
;  1868        U 1864  2              .BUFFER[HM2$W_ALTIDXVBN] NEQ .CLUSTER*3+1 OR
;  1869        U 1865  2              .BUFFER[HM2$W_IBMAPVBN] NEQ .CLUSTER*4+1 OR
;  1870        U 1866  2              CHECK_LBN(.BUFFER[HM2$L_IBMAPLBN], .CLUSTER*4+1) OR
```

```
;  1871        U 1867  2          .BUFFER[HM2$L_MAXFILES] LEQU .BUFFER[HM2$W_RESFILES] OR
;  1872        U 1868  2          .BUFFER[HM2$W_IBMAPSIZE] NEQ (.BUFFER[HM2$L_MAXFILES] + 4095) / 4096 OR
;  1873        U 1869  2          .BUFFER[HM2$W_RESFILES] LSSU 5 OR
;  1874        U 1870  2          (IF .VOLUME_COUNT GTR 1
;  1875        U 1871  2          THEN
;  1876        U 1872  2              .BUFFER[HM2$W_RVN] NEQ .RVN
;  1877        U 1873  2              ! strucname
;  1878        U 1874  2          ELSE
;  1879        U 1875  2              .BUFFER[HM2$W_RVN] NEQ 0 OR
;  1880        U 1876  2              CH$FIND_NOT_CH(HM2$S_STRUCNAME, BUFFER[HM2$T_STRUCNAME], %C' ')
;  1881        U 1877  2                  NEQ 0) OR
;  1882        U 1878  2          (IF .BUFFER[HM2$W_RVN] EQL 1
;  1883        U 1879  2          THEN
;  1884        U 1880  2              .BUFFER[HM2$W_SETCOUNT] LEQU 1
;  1885        U 1881  2          ELSE
;  1886        U 1882  2              .BUFFER[HM2$W_SETCOUNT] NEQ 0) OR
;  1887        U 1883  2          ! credate
;  1888        U 1884  2          ! volname
;  1889        U 1885  2          ! ownername
;  1890        U 1886  2          CH$NEQ(
;  1891        U 1887  2              HM2$S_FORMAT, BUFFER[HM2$T_FORMAT],
;  1892        U 1888  2              HM2$S_FORMAT, UPLIT BYTE ('DECFILE11B  '))
;  1893        U 1889  2          END
;  1894        U 1890  2      ELSE
;  1895        U 1891  2          BEGIN
;  1896        U 1892  2          !
;  1897        U 1893  2          ! The following fields have already been verified:
;  1898        U 1894  2          !       HM1$W_MAXFILES, HM1$W_STRUCLEV,
;  1899        U 1895  2          !       HM1$W_CHECKSUM1, HM1$W_CHECKSUM2.
;  1900        U 1896  2          !
;  1901        U 1897  2          ! The following fields are considered entirely free values:
;  1902        U 1898  2          !       HM1$W_DEVTYPE, HM1$W_VOLOWNER, HM1$W_PROTECT, HM1$W_VOLCHAR,
;  1903        U 1899  2          !       HM1$W_FILEPROT, HM1$B_WINDOW, HM1$B_EXTEND, HM1$B_LRU_LIM,
;  1904        U 1900  2          !       HM1$L_SERIALNUM.
;  1905        U 1901  2          !
;  1906        U 1902  2          .BUFFER[HM1$W_IBMAPSIZE] NEQ (.BUFFER[HM1$W_MAXFILES] + 4095) / 4096 OR
;  1907        U 1903  2          CHECK_LBN(ROT(.BUFFER[HM1$L_IBMAPLBN], 16), 3) OR
;  1908        U 1904  2          .BUFFER[HM1$W_CLUSTER] NEQ 1 OR
;  1909        U 1905  2          ! volname
;  1910        U 1906  2          ! credate
;  1911        U 1907  2          ! volname2
;  1912        U 1908  2          ! ownername
;  1913        U 1909  2          CH$NEQ(
;  1914        U 1910  2              HM1$S_FORMAT, BUFFER[HM1$T_FORMAT],
;  1915        U 1911  2              HM1$S_FORMAT, UPLIT BYTE ('DECFILE11A  '))
;  1916        U 1912  2          END
;  1917        U 1913  2      END
;  1918        U 1914  2  THEN
;  1919        U 1915  2      BEGIN
;  1920        U 1916  2      SIGNAL(VERIFY$_CHKPRIHOME, 2, .HOMEVBN, .RVN);
;  1921        U 1917  2      END;
;  1922        U 1918  2
;  1923        U 1919  2
;  1924          1920  2  %fi
;  1925          1921  2
;  1926          1922  2
;  1927          1923  2  ! Read the boot block to find out if it is good.
```

```
;  1928          1924  2 !
;  1929        P 1925  2 STATUS = $QIOW(
;  1930        P 1926  2     FUNC=IO$_READVBLK,
;  1931        P 1927  2     CHAN=.CHANNEL,
;  1932        P 1928  2     IOSB=IOSB,
;  1933        P 1929  2     P1=HDR_BUFFER_2,
;  1934        P 1930  2     P2=512,
;  1935          1931  2     P3=1);
;  1936          1932  2 IF .STATUS THEN STATUS = .IOSB[0];
;  1937          1933  2 IF NOT .STATUS
;  1938          1934  2 THEN
;  1939          1935  2     SIGNAL(VERIFY$_READBOOT, 1, .RVN, .STATUS);
;  1940          1936  2
;  1941          1937  2
;  1942          1938  2 ! Check all of the home blocks.
;  1943          1939  2 !
;  1944          1940  2 INCR VBN FROM 2 TO (IF .STRUCTURE_LEVEL EQL 2 THEN 3*.CLUSTER ELSE 2) DO
;  1945          1941  3     BEGIN
;  1946          1942  3     IF .VBN NEQ .HOMEVBN
;  1947          1943  3     THEN
;  1948          1944  4         BEGIN
;  1949          1945  4
;  1950          1946  4         ! Read the block.
;  1951          1947  4         !
;  1952        P 1948  4         STATUS = $QIOW(
;  1953        P 1949  4             FUNC=IO$_READVBLK,
;  1954        P 1950  4             CHAN=.CHANNEL,
;  1955        P 1951  4             IOSB=IOSB,
;  1956        P 1952  4             P1=HDR_BUFFER_2,
;  1957        P 1953  4             P2=512,
;  1958          1954  4             P3=.VBN);
;  1959          1955  4         IF .STATUS THEN STATUS = .IOSB[0];
;  1960          1956  4
;  1961          1957  4
;  1962          1958  4         ! Check the validity of the block.  If it reads with an error, always
;  1963          1959  4         ! consider it invalid so that it will be rewritten.  Otherwise, compare
;  1964          1960  4         ! for equality to the primary home block, verifying the expected
;  1965          1961  4         ! differences.
;  1966          1962  4         !
;  1967          1963  4         IF
;  1968          1964  5             BEGIN
;  1969          1965  5             IF NOT .STATUS
;  1970          1966  5             THEN
;  1971          1967  5                 TRUE
;  1972          1968  5             ELSE IF .STRUCTURE_LEVEL EQL 2
;  1973          1969  5             THEN
;  1974          1970  6                 BEGIN
;  1975          1971  6                 IF
;  1976          1972  7                     BEGIN
;  1977          1973  7                     NOT CHECKSUM2(HDR_BUFFER_2, $BYTEOFFSET(HM2$W_CHECKSUM1)) OR
;  1978          1974  7                     NOT CHECKSUM2(HDR_BUFFER_2, $BYTEOFFSET(HM2$W_CHECKSUM2)) OR
;  1979          1975  7                     CHECK_LBN(.HDR_BUFFER_2[HM2$L_HOMELBN], .VBN) OR
;  1980          1976  7                     .HDR_BUFFER_2[HM2$W_HOMEVBN] NEQ .VBN
;  1981          1977  7                     END
;  1982          1978  6                 THEN
;  1983          1979  6                     TRUE
;  1984          1980  6                 ELSE
```

```
;  1985     1981  7                    BEGIN
;  1986     1982  7                    BUFFER[HM2$L_HOMELBN] = 0;
;  1987     1983  7                    BUFFER[HM2$W_HOMEVBN] = 0;
;  1988     1984  7                    BUFFER[HM2$W_CHECKSUM1] = 0;
;  1989     1985  7                    BUFFER[HM2$W_CHECKSUM2] = 0;
;  1990     1986  7                    HDR_BUFFER_2[HM2$L_HOMELBN] = 0;
;  1991     1987  7                    HDR_BUFFER_2[HM2$W_HOMEVBN] = 0;
;  1992     1988  7                    HDR_BUFFER_2[HM2$W_CHECKSUM1] = 0;
;  1993     1989  7                    HDR_BUFFER_2[HM2$W_CHECKSUM2] = 0;
;  1994     1990  7                    CH$NEQ(512, HDR_BUFFER_2, 512, BUFFER)
;  1995     1991  7                    END
;  1996     1992  6                END
;  1997     1993  5            ELSE
;  1998     1994  5                CH$NEQ(512, HDR_BUFFER_2, 512, BUFFER)
;  1999     1995  5            END
;  2000     1996  4        THEN
;  2001     1997  5            BEGIN
;  2002     1998  5
;  2003     1999  5            ! Report an appropriate error.
;  2004     2000  5            !
;  2005     2001  5            IF .STATUS
;  2006     2002  5                THEN SIGNAL(VERIFY$_CHKALTHOME, 2, .VBN, .RVN)
;  2007     2003  5                ELSE SIGNAL(VERIFY$_READHOME, 2, .VBN, .RVN, .STATUS);
;  2008     2004  5
;  2009     2005  5
;  2010     2006  5            ! Reconstruct and rewrite the block.
;  2011     2007  5            !
;  2012     2008  5            IF DO_REPAIR()
;  2013     2009  5            THEN
;  2014     2010  6                BEGIN
;  2015     2011  6                CH$MOVE(512, BUFFER, HDR_BUFFER_2);
;  2016     2012  6                IF .STRUCTURE_LEVEL EQL 2
;  2017     2013  6                THEN
;  2018     2014  7                    BEGIN
;  2019     2015  7                    HDR_BUFFER_2[HM2$W_HOMEVBN] = .VBN;
;  2020     2016  7                    MAP_VIRTUA[(.WINDOW, .VBN, HDR_BUFFER_2[HM2$L_HOMELBN]);
;  2021     2017  7                    CHECKSUM2(HDR_BUFFER_2, $BYTEOFFSET(HM2$W_CHECKSUM1));
;  2022     2018  7                    CHECKSUM2(HDR_BUFFER_2, $BYTEOFFSET(HM2$W_CHECKSUM2));
;  2023     2019  6                    END;
;  2024   P 2020  6                STATUS = $QIOW(
;  2025   P 2021  6                    FUNC=IO$_WRITEVBLK,
;  2026   P 2022  6                    CHAN=.CHANNEL,
;  2027   P 2023  6                    IOSB=IOSB,
;  2028   P 2024  6                    P1=HDR_BUFFER_2,
;  2029   P 2025  6                    P2=512,
;  2030     2026  6                    P3=.VBN);
;  2031     2027  6                IF .STATUS THEN STATUS = .IOSB[0];
;  2032     2028  6                IF NOT .STATUS
;  2033     2029  6                THEN
;  2034     2030  6                    SIGNAL(VERIFY$_WRITEHOME, 2, .VBN, .RVN, .STATUS)
;  2035     2031  5                END;
;  2036     2032  4            END;
;  2037     2033  3        END;
;  2038     2034  2    END;
;  2039     2035  2
;  2040     2036  2
;  2041     2037  2 ! Allocate memory for and read index file bitmap.
```

```
;  2042     2038  2 !
;  2043     2039  2 STATUS = LIB$GET_VM(%REF(.IMAP_SIZE[.RVN-1] * 512), IMAP[.RVN-1]);
;  2044     2040  2 IF NOT .STATUS THEN SIGNAL(VERIFY$_ALLOCMEM, 0,.STATUS);
;  2045     2041  2 INCR VBN FROM 0 TO .IMAP_SIZE[.RVN-1] - 1 BY 127 DO
;  2046     2042  2     BEGIN
;  2047     2043  2     LOCAL
;  2048     2044  2         THIS_BLOCKS;    ! Count of blocks to read on current iteration
;  2049     2045  2
;  2050     2046  2
;  2051     2047  2     ! Compute number of blocks to read this time.
;  2052     2048  2     !
;  2053     2049  2     THIS_BLOCKS = MINU(
;  2054     2050  2         127,
;  2055     2051  2         .IMAP_SIZE[.RVN-1] - .VBN);
;  2056     2052  2
;  2057     2053  2
;  2058     2054  2     ! Read the blocks.  If this fails, re-execute the read one block
;  2059     2055  2     ! at a time noting the blocks that fail.
;  2060     2056  2     !
;  2061   P 2057  2     STATUS = $QIOW(
;  2062   P 2058  2         FUNC=IO$_READVBLK,
;  2063   P 2059  2         CHAN=.CHANNEL,
;  2064   P 2060  2         IOSB=IOSB,
;  2065   P 2061  2         P1=.IMAP[.RVN-1] + .VBN * 512,
;  2066   P 2062  2         P2=.THIS_BLOCKS * 512,
;  2067     2063  2         P3=.BITMAP_OFFSET[.RVN-1] + .VBN);
;  2068     2064  2     IF .STATUS THEN STATUS = .IOSB[0];
;  2069     2065  2     IF NOT .STATUS
;  2070     2066  2     THEN
;  2071     2067  4         BEGIN
;  2072     2068  4         INCR XVBN FROM 0 TO .THIS_BLOCKS-1 DO
;  2073     2069  5             BEGIN
;  2074   P 2070  5             STATUS = $QIOW(
;  2075   P 2071  5                 FUNC=IO$_READVBLK,
;  2076   P 2072  5                 CHAN=.CHANNEL,
;  2077   P 2073  5                 IOSB=IOSB,
;  2078   P 2074  5                 P1=.IMAP[.RVN-1] + .VBN * 512 + .XVBN * 512,
;  2079   P 2075  5                 P2=512,
;  2080     2076  5                 P3=.BITMAP_OFFSET[.RVN-1] + .VBN + .XVBN);
;  2081     2077  5             IF .STATUS THEN STATUS = .IOSB[0];
;  2082     2078  5             IF NOT .STATUS
;  2083     2079  5             THEN
;  2084     2080  5                 SIGNAL(
;  2085     2081  5                     VERIFY$_READIBMAP,
;  2086     2082  5                     2,
;  2087     2083  5                     .BITMAP_OFFSET[.RVN-1] + .VBN + .XVBN,
;  2088     2084  5                     .RVN,
;  2089     2085  5                     .STATUS);
;  2090     2086  4             END;
;  2091     2087  3         END;
;  2092     2088  2     END;
;  2093     2089  2
;  2094     2090  2
;  2095     2091  2 ! Get the EOF from the index file header.
;  2096     2092  2 !
;  2097     2093  2 EOF[.RVN-1] = 0;
;  2098     2094  2 IF .STRUCTURE_LEVEL EQL 2
```

```
;  2099     2095  2 THEN
;  2100     2096  2     EOF[.RVN-1] = ROT(.BBLOCK[HDR_BUFFER[FH2$W_RECATTR], FAT$L_EFBLK], 16) - 1;
;  2101     2097  2
;  2102     2098  2
;  2103     2099  2 ! Now scan the volume's index file bitmap backwards, looking for the highest
;  2104     2100  2 ! bit set.  The maximum of the EOF mark and the highest set bit is taken to
;  2105     2101  2 ! be the true index file EOF.
;  2106     2102  2 !
;  2107     2103  2 DECR J FROM .IMAP_SIZE[.RVN-1] * 128 - 1 TO 0 DO
;  2108     2104  3     BEGIN
;  2109     2105  3     IF .VECTOR[.IMAP[.RVN-1], .J] NEQ 0
;  2110     2106  3     THEN
;  2111     2107  4         BEGIN
;  2112     2108  4         EOF[.RVN-1] = MAXU(
;  2113     2109  4                 .J*32 +
;  2114     2110  4                 LEFT_ONE(.VECTOR[.IMAP[.RVN-1], .J]) +
;  2115     2111  4                 .HEADER_OFFSET[.RVN-1],
;  2116     2112  4             .EOF[.RVN-1]);
;  2117     2113  4         EXITLOOP;
;  2118     2114  4         END;
;  2119     2115  3     END;
;  2120     2116  2
;  2121     2117  2
;  2122     2118  2 ! Allocate memory for and initialize directory bitmap.
;  2123     2119  2 !
;  2124     2120  2 STATUS = LIB$GET_VM(%REF(.IMAP_SIZE[.RVN-1] * 512), DIRMAP[.RVN-1]);
;  2125     2121  2 IF NOT .STATUS THEN SIGNAL(VERIFY$_ALLOCMEM, 0, .STATUS);
;  2126     2122  2 INITIALIZE(0, .IMAP_SIZE[.RVN-1] * 128, .DIRMAP[.RVN-1]);
;  2127     2123  2
;  2128     2124  2
;  2129     2125  2 ! Allocate memory for and initialize lost file bitmap.
;  2130     2126  2 !
;  2131     2127  2 STATUS = LIB$GET_VM(%REF(.IMAP_SIZE[.RVN-1] * 512), LOSTMAP[.RVN-1]);
;  2132     2128  2 IF NOT .STATUS THEN SIGNAL(VERIFY$_ALLOCMEM, 0, .STATUS);
;  2133     2129  2 INITIALIZE(0, .IMAP_SIZE[.RVN-1] * 128, .LOSTMAP[.RVN-1]);
;  2134     2130  2
;  2135     2131  2
;  2136     2132  2 ! Allocate memory for and initialize extension header bitmap.
;  2137     2133  2 !
;  2138     2134  2 STATUS = LIB$GET_VM(%REF(.IMAP_SIZE[.RVN-1] * 512), EXTMAP[.RVN-1]);
;  2139     2135  2 IF NOT .STATUS THEN SIGNAL(VERIFY$_ALLOCMEM, 0, .STATUS);
;  2140     2136  2 INITIALIZE(0, .IMAP_SIZE[.RVN-1] * 128, .EXTMAP[.RVN-1]);
;  2141     2137  2
;  2142     2138  2
;  2143     2139  2 ! Allocate memory for file sequence number vector.  It need not be
;  2144     2140  2 ! initialized -- an entry is valid only if the corresponding IMAP
;  2145     2141  2 ! bit is set.
;  2146     2142  2 !
;  2147     2143  2 STATUS = LIB$GET_VM(%REF(.IMAP_SIZE[.RVN-1] * 512 * 16), SEQMAP[.RVN-1]);
;  2148     2144  2 IF NOT .STATUS THEN SIGNAL(VERIFY$_ALLOCMEM, 0, .STATUS);
;  2149     2145  2
;  2150     2146  2
;  2151     2147  2 IF .QUAL[QUAL_USAG]
;  2152     2148  2 THEN
;  2153     2149  3     BEGIN
;  2154     2150  3
;  2155     2151  3     ! Allocate memory for file owner UIC vector.  It need not be
```

```
;  2156     2152  3     ! initialized -- an entry is valid only if the corresponding IMAP
;  2157     2153  3     ! bit is set.
;  2158     2154  3     !
;  2159     2155  3     STATUS = LIB$GET_VM(%REF(.IMAP_SIZE[.RVN-1] * 512 * 32), OWNER[.RVN-1]);
;  2160     2156  3     IF NOT .STATUS THEN SIGNAL(VERIFY$_ALLOCMEM, 0, .STATUS);
;  2161     2157  3
;  2162     2158  3
;  2163     2159  3     ! Allocate memory for file allocated blocks vector.  It need not be
;  2164     2160  3     ! initialized -- an entry is valid only if the corresponding IMAP
;  2165     2161  3     ! bit is set.
;  2166     2162  3     !
;  2167     2163  3     STATUS = LIB$GET_VM(%REF(.IMAP_SIZE[.RVN-1] * 512 * 32), ALLOCATION[.RVN-1]);
;  2168     2164  3     IF NOT .STATUS THEN SIGNAL(VERIFY$_ALLOCMEM, 0, .STATUS);
;  2169     2165  3
;  2170     2166  3
;  2171     2167  3     ! Allocate memory for file used blocks vector.  It need not be
;  2172     2168  3     ! initialized -- an entry is valid only if the corresponding IMAP
;  2173     2169  3     ! bit is set.
;  2174     2170  3     !
;  2175     2171  3     STATUS = LIB$GET_VM(%REF(.IMAP_SIZE[.RVN-1] * 512 * 32), USAGE[.RVN-1]);
;  2176     2172  3     IF NOT .STATUS THEN SIGNAL(VERIFY$_ALLOCMEM, 0, .STATUS);
;  2177     2173  2     END;
;  2178     2174  2
;  2179     2175  2
;  2180     2176  2 IF .STRUCTURE_LEVEL EQL 2
;  2181     2177  2 THEN
;  2182     2178  3     BEGIN
;  2183     2179  3
;  2184     2180  3     ! Allocate memory for file back link FID vector.  It need not be
;  2185     2181  3     ! initialized -- an entry is valid only if the corresponding IMAP
;  2186     2182  3     ! bit is set.
;  2187     2183  3     !
;  2188     2184  3     STATUS = LIB$GET_VM(%REF(.IMAP_SIZE[.RVN-1] * 512 * 48), BACKMAP[.RVN-1]);
;  2189     2185  3     IF NOT .STATUS THEN SIGNAL(VERIFY$_ALLOCMEM, 0, .STATUS);
;  2190     2186  3
;  2191     2187  3
;  2192     2188  3     ! Read the primary index file header into HDR_BUFFER.  If the read fails,
;  2193     2189  3     ! force the header invalid so that it will be rewritten.
;  2194     2190  3     !
;  2195   P 2191  3     STATUS = $QIOW(
;  2196   P 2192  3         FUNC=IO$_READVBLK,
;  2197   P 2193  3         CHAN=.CHANNEL,
;  2198   P 2194  3         IOSB=IOSB,
;  2199   P 2195  3         P1=HDR_BUFFER,
;  2200   P 2196  3         P2=512,
;  2201     2197  3         P3=.HEADER_OFFSET[.RVN-1] + FID$C_INDEXF);
;  2202     2198  3     IF .STATUS THEN STATUS = .IOSB[0];
;  2203     2199  3     IF NOT .STATUS
;  2204     2200  3     THEN
;  2205     2201  4         BEGIN
;  2206     2202  4         HEADER_ERROR(VERIFY$_READHEADER, INDEX_FILE_ID, HDR_BUFFER, .STATUS);
;  2207     2203  4         HDR_BUFFER[FH2$B_STRUCLEV] = 0;
;  2208     2204  4         END;
;  2209     2205  3
;  2210     2206  3
;  2211     2207  3     ! Read the alternate index file header into HDR_BUFFER_2.  If the read
;  2212     2208  3     ! fails, force the header invalid so that it will be rewritten.
```

```
;  2213          2209  3     !
;  2214        P 2210  3     STATUS = $QIOW(
;  2215        P 2211  3         FUNC=IO$_READVBLK,
;  2216        P 2212  3         CHAN=.CHANNEL,
;  2217        P 2213  3         IOSB=IOSB,
;  2218        P 2214  3         P1=HDR_BUFFER_2,
;  2219        P 2215  3         P2=512,
;  2220          2216  3         P3=.CLUSTER*3 + 1);
;  2221          2217  3     IF .STATUS THEN STATUS = .IOSB[0];
;  2222          2218  3     IF NOT .STATUS
;  2223          2219  3     THEN
;  2224          2220  4         BEGIN
;  2225          2221  4         HEADER_ERROR(VERIFY$_READHEADER, INDEX_FILE_ID, HDR_BUFFER_2, .STATUS);
;  2226          2222  4         HDR_BUFFER_2[FH2$B_STRUCLEV] = 0;
;  2227          2223  4         END;
;  2228          2224  3
;  2229          2225  3
;  2230          2226  3     ! Check condition of primary and alternate index file headers.
;  2231          2227  3     !
;  2232          2228  3     IF VERIFY_HEADER(HDR_BUFFER, INDEX_FILE_ID)
;  2233          2229  3     THEN
;  2234          2230  4         BEGIN
;  2235          2231  4         IF
;  2236          2232  5             BEGIN
;  2237          2233  5             IF NOT VERIFY_HEADER(HDR_BUFFER_2, INDEX_FILE_ID)
;  2238          2234  5             THEN
;  2239          2235  5                 TRUE
;  2240          2236  5             ELSE
;  2241          2237  5                 .HDR_BUFFER[FH2$B_MAP_INUSE] NEQ .HDR_BUFFER_2[FH2$B_MAP_INUSE] OR
;  2242          2238  5                 CH$NEQ(
;  2243          2239  5                     .HDR_BUFFER[FH2$B_ACOFFSET] - .HDR_BUFFER[FH2$B_MPOFFSET],
;  2244          2240  5                     HDR_BUFFER + .HDR_BUFFER[FH2$B_MPOFFSET] * 2,
;  2245          2241  5                     .HDR_BUFFER_2[FH2$B_ACOFFSET] - .HDR_BUFFER_2[FH2$B_MPOFFSET],
;  2246          2242  5                     HDR_BUFFER_2 + .HDR_BUFFER_2[FH2$B_MPOFFSET] * 2) OR
;  2247          2243  5                 .BBLOCK[HDR_BUFFER[FH2$Q_RECATTR], FAT$L_EFBLK] NEQ .BBLOCK[HDR_BUFFER_2[FH2$W_RECATTR], FAT
;  2248          2244  5             END
;  2249          2245  4         THEN
;  2250          2246  5             BEGIN
;  2251          2247  5             !
;  2252          2248  5             ! Primary header good.  Alternate header fails basic validity or
;  2253          2249  5             ! has a different EFBLK or map area.  Alternate header will be
;  2254          2250  5             ! restored from primary.
;  2255          2251  5             !
;  2256          2252  5             SIGNAL(VERIFY$_ALTIHDBAD, 1, .RVN);
;  2257          2253  5             IF DO_REPAIR()
;  2258          2254  5             THEN
;  2259          2255  6                 BEGIN
;  2260        P 2256  6                 STATUS = $QIOW(
;  2261        P 2257  6                     FUNC=IO$_WRITEVBLK,
;  2262        P 2258  6                     CHAN=.CHANNEL,
;  2263        P 2259  6                     IOSB=IOSB,
;  2264        P 2260  6                     P1=HDR_BUFFER,
;  2265        P 2261  6                     P2=512,
;  2266          2262  6                     P3=.CLUSTER*3 + 1);
;  2267          2263  6                 IF .STATUS THEN STATUS = .IOSB[0];
;  2268          2264  6                 IF NOT .STATUS
;  2269          2265  6                 THEN
```

```
;  2270          2266  6                     HEADER_ERROR(
;  2271          2267  6                         VERIFY$_WRITEHEADER, INDEX_FILE_ID, HDR_BUFFER,
;  2272          2268  6                         .STATUS);
;  2273          2269  5                 END;
;  2274          2270  5             END
;  2275          2271  4         END
;  2276          2272  3     ELSE
;  2277          2273  4         BEGIN
;  2278          2274  4         IF VERIFY_HEADER(HDR_BUFFER_2, INDEX_FILE_ID)
;  2279          2275  4         THEN
;  2280          2276  5             BEGIN
;  2281          2277  5             !
;  2282          2278  5             ! Primary header is bad, alternate header good.
;  2283          2279  5             ! Primary header will be restored from alternate.
;  2284          2280  5             !
;  2285          2281  5             SIGNAL(VERIFY$_PRIIHDBAD, 1, .RVN);
;  2286          2282  5             IF DO_REPAIR()
;  2287          2283  5             THEN
;  2288          2284  6                 BEGIN
;  2289        P 2285  6                 STATUS = $QIOW(
;  2290        P 2286  6                     FUNC=IO$_WRITEVBLK,
;  2291        P 2287  6                     CHAN=.CHANNEL,
;  2292        P 2288  6                     IOSB=IOSB,
;  2293        P 2289  6                     P1=HDR_BUFFER_2,
;  2294        P 2290  6                     P2=512,
;  2295          2291  6                     P3=.HEADER_OFFSET[.RVN-1] + FID$C_INDEXF);
;  2296          2292  6                 IF .STATUS THEN STATUS = .IOSB[0];
;  2297          2293  6                 IF NOT .STATUS
;  2298          2294  6                 THEN
;  2299          2295  6                     HEADER_ERROR(
;  2300          2296  6                         VERIFY$_WRITEHEADER, INDEX_FILE_ID, HDR_BUFFER_2,
;  2301          2297  6                         .STATUS);
;  2302          2298  5                 END;
;  2303          2299  5             END
;  2304          2300  4         ELSE
;  2305          2301  4             SIGNAL(VERIFY$_FINDIHD, 1, .RVN);
;  2306          2302  3         END;
;  2307          2303  2     END;
;  2308          2304  2
;  2309          2305  2
;  2310          2306  2 ! Delete the index file window.
;  2311          2307  2 !
;  2312          2308  2 DELETE_WINDOW(.WINDOW);
;  2313          2309  2
;  2314          2310  2
;  2315          2311  2 ! Deaccess the index file.
;  2316          2312  2 !
;  2317        P 2313  2 $QIOW(
;  2318        P 2314  2     FUNC=IO$_DEACCESS,
;  2319          2315  2     CHAN=.CHANNEL);
;  2320          2316  2
;  2321          2317  2
;  2322          2318  2 ! Access the storage bitmap file.  Read the file header into HDR_BUFFER.
;  2323          2319  2 !
;  2324          2320  2 CH$FILL(0, FIB$C_LENGTH, FIB);
;  2325          2321  2 FIB[FIB$L_ACCTL] = .ACCTL_VALUE;
;  2326          2322  2 FIB[FIB$W_FID_NUM] = FID$C_BITMAP;
```

```
;  2327          2323  2 FIB[FIB$W_FID_SEQ] = FID$C_BITMAP;
;  2328          2324  2 FIB[FIB$W_FID_RVN] = .RVN;
;  2329        P 2325  2 STATUS = $QIOW(
;  2330        P 2326  2     FUNC=IO$_ACCESS OR IO$M_ACCESS,
;  2331        P 2327  2     CHAN=.CHANNEL,
;  2332        P 2328  2     IOSB=IOSB,
;  2333        P 2329  2     P1=FIB_DESC,
;  2334          2330  2     P5=HDR_ATR_DESC);
;  2335          2331  2 IF .STATUS THEN STATUS = .IOSB[0];
;  2336          2332  2 IF NOT .STATUS
;  2337          2333  2 THEN
;  2338          2334  2     SIGNAL(VERIFY$_OPENBITMAP, 1, .RVN, .STATUS);
;  2339          2335  2
;  2340          2336  2
;  2341          2337  2 ! Get a window block for the bitmap file.
;  2342          2338  2 !
;  2343          2339  2 WINDOW = CREATE_WINDOW(HDR_BUFFER, .RVN);
;  2344          2340  2
;  2345          2341  2
;  2346          2342  2 ! Read the storage control block.
;  2347          2343  2 !
;  2348        P 2344  2 STATUS = $QIOW(
;  2349        P 2345  2     FUNC=IO$_READVBLK,
;  2350        P 2346  2     CHAN=.CHANNEL,
;  2351        P 2347  2     IOSB=IOSB,
;  2352        P 2348  2     P1=BUFFER,
;  2353        P 2349  2     P2=512,
;  2354          2350  2     P3=1);
;  2355          2351  2 IF .STATUS THEN STATUS = .IOSB[0];
;  2356          2352  2 IF NOT .STATUS
;  2357          2353  2 THEN
;  2358          2354  2     SIGNAL(VERIFY$_READSCB, 1, .RVN, .STATUS);
;  2359          2355  2
;  2360          2356  2
;  2361          2357  2 ! Validate the storage control block.
;  2362          2358  2 !
;  2363          2359  2 IF
;  2364          2360  3     BEGIN
;  2365          2361  3     IF NOT .STATUS
;  2366          2362  3     THEN
;  2367          2363  3         TRUE
;  2368          2364  3     ELSE IF .STRUCTURE_LEVEL EQL 2
;  2369          2365  3     THEN
;  2370          2366  4         BEGIN
;  2371          2367  4         NOT CHECKSUM(BUFFER) OR
;  2372          2368  4         .BUFFER[SCB$W_STRUCLEV] NEQ SCB$C_LEVEL2+1 OR
;  2373          2369  4         .BUFFER[SCB$W_CLUSTER] NEQ .CLUSTER OR
;  2374          2370  4         .BUFFER[SCB$L_VOLSIZE] NEQ .DEVICE_CHAR[DVI_MAXBLOCK] OR
;  2375          2371  4         .BUFFER[SCB$L_BLKSIZE] NEQ
;  2376          2372  5             (.DEVICE_CHAR[DVI_SECTORS] * .DEVICE_CHAR[DVI_TRACKS] *
;  2377          2373  4             .DEVICE_CHAR[DVI_CYLINDERS]) / .DEVICE_CHAR[DVI_MAXBLOCK] OR
;  2378          2374  4         .BUFFER[SCB$L_SECTORS] NEQ .DEVICE_CHAR[DVI_SECTORS] OR
;  2379          2375  4         .BUFFER[SCB$L_TRACKS] NEQ .DEVICE_CHAR[DVI_TRACKS] OR
;  2380          2376  4         .BUFFER[SCB$L_CYLINDER] NEQ .DEVICE_CHAR[DVI_CYLINDERS]
;  2381          2377  4         END
;  2382          2378  3     ELSE
;  2383          2379  4         BEGIN
```

```
;  2384          2380  4          MAP
;  2385          2381  4              BUFFER:             VECTOR;
;  2386          2382  4          LOCAL
;  2387          2383  4              BLOCK_COUNT;
;  2388          2384  4
;  2389          2385  4          BLOCK_COUNT = .SMAP_SIZE[.RVN-1];
;  2390          2386  4          IF .BLOCK_COUNT GTRU 126 THEN BLOCK_COUNT = 0;
;  2391          2387  4          .(BUFFER+3)<0,8> NEQ .BLOCK_COUNT OR
;  2392          2388  4          .BUFFER[.BLOCK_COUNT+1] NEQ ROT(.DEVICE_CHAR[DVI_MAXBLOCK], 16)
;  2393          2389  4          END
;  2394          2390  3      END
;  2395          2391  2  THEN
;  2396          2392  3      BEGIN
;  2397          2393  3      IF .STATUS THEN SIGNAL(VERIFY$_CHKSCB, 1, .RVN);
;  2398          2394  3      IF DO_REPAIR()
;  2399          2395  3      THEN
;  2400          2396  4          BEGIN
;  2401          2397  4
;  2402          2398  4          ! Reconstruct the storage control block.
;  2403          2399  4          !
;  2404          2400  4          CH$FILL(0, 512, BUFFER);
;  2405          2401  4          IF .STRUCTURE_LEVEL EQL 2
;  2406          2402  4          THEN
;  2407          2403  5              BEGIN
;  2408          2404  5              BUFFER[SCB$W_STRUCLEV] = SCB$C_LEVEL2+1;
;  2409          2405  5              BUFFER[SCB$W_CLUSTER] = .CLUSTER;
;  2410          2406  5              BUFFER[SCB$L_VOLSIZE] = .DEVICE_CHAR[DVI_MAXBLOCK];
;  2411          2407  5              BUFFER[SCB$L_BLKSIZE] =
;  2412          2408  6                  (.DEVICE_CHAR[DVI_SECTORS] * .DEVICE_CHAR[DVI_TRACKS] *
;  2413          2409  5                  .DEVICE_CHAR[DVI_CYLINDERS]) / .DEVICE_CHAR[DVI_MAXBLOCK];
;  2414          2410  5              BUFFER[SCB$L_SECTORS] = .DEVICE_CHAR[DVI_SECTORS];
;  2415          2411  5              BUFFER[SCB$L_TRACKS] = .DEVICE_CHAR[DVI_TRACKS];
;  2416          2412  5              BUFFER[SCB$L_CYLINDER] = .DEVICE_CHAR[DVI_CYLINDERS];
;  2417          2413  5              BUFFER[SCB$L_STATUS] =
;  2418          2414  5                  SCB$M_MAPALLOC OR SCB$M_FILALLOC OR SCB$M_HDRWRITE;
;  2419          2415  5              CHECKSUM(BUFFER);
;  2420          2416  5              END
;  2421          2417  4          ELSE
;  2422          2418  5              BEGIN
;  2423          2419  5              MAP
;  2424          2420  5                  BUFFER:         VECTOR;
;  2425          2421  5              LOCAL
;  2426          2422  5                  BLOCK_COUNT;
;  2427          2423  5
;  2428          2424  5              BLOCK_COUNT = .SMAP_SIZE[.RVN-1];
;  2429          2425  5              IF .BLOCK_COUNT GTRU 126 THEN BLOCK_COUNT = 0;
;  2430          2426  5              (BUFFER+3)<0,8> = .BLOCK_COUNT;
;  2431          2427  5              INCR J FROM 0 TO .BLOCK_COUNT-1 DO BUFFER[.J+1] = 4096;
;  2432          2428  5              BUFFER[.BLOCK_COUNT+1] = ROT(.DEVICE_CHAR[DVI_MAXBLOCK], 16);
;  2433          2429  4              END;
;  2434          2430  4
;  2435          2431  4
;  2436          2432  4          ! Rewrite the storage control block.
;  2437          2433  4          !
;  2438        P 2434  4          STATUS = $QIOW(
;  2439        P 2435  4              FUNC=IO$_WRITEVBLK,
;  2440        P 2436  4              CHAN=.CHANNEL,
```

```
;  2441        P 2437  4              IOSB=IOSB,
;  2442        P 2438  4              P1=BUFFER,
;  2443        P 2439  4              P2=512,
;  2444          2440  4              P3=1);
;  2445          2441  4          IF .STATUS THEN STATUS = .IOSB[0];
;  2446          2442  4          IF NOT .STATUS
;  2447          2443  4          THEN
;  2448          2444  4              SIGNAL(VERIFY$_WRITESCB, 1, .RVN, .STATUS);
;  2449          2445  4          END;
;  2450          2446  3      END;
;  2451          2447  3
;  2452          2448  3
;  2453          2449  3  ! Check bitmap file for contiguity and correct size.
;  2454          2450  3  !
;  2455          2451  3  IF
;  2456          2452  3      BEGIN
;  2457          2453  3      IF .WINDOW EQL 0
;  2458          2454  3      THEN
;  2459          2455  3          TRUE
;  2460          2456  3      ELSE
;  2461          2457  3          .WINDOW[WDW_SIZE] NEQ 1 OR
;  2462          2458  3          .BBLOCK[WINDOW[WDW_ENTRY], WDW_COUNT] LSSU .SMAP_SIZE[.RVN-1] + 1
;  2463          2459  3      END
;  2464          2460  3  THEN
;  2465          2461  3      SIGNAL(VERIFY$_BADBITMAP, 1, .RVN);
;  2466          2462  3
;  2467          2463  3
;  2468          2464  2  ! Allocate memory for recomputed storage bitmap and initialize it.
;  2469          2465  2  ! (Set bits mean free clusters.)  Mark clusters above the true size of
;  2470          2466  2  ! the volume allocated.
;  2471          2467  2  !
;  2472          2468  2  STATUS = LIB$GET_VM(%REF(.SMAP_SIZE[.RVN-1] * 512), NSMAP[.RVN-1]);
;  2473          2469  2  IF NOT .STATUS THEN SIGNAL(VERIFY$_ALLOCMEM, 0, .STATUS);
;  2474          2470  2  INITIALIZE(-1, .SMAP_SIZE[.RVN-1] * 128, .NSMAP[.RVN-1]);
;  2475          2471  2  INCR N
;  2476          2472  3      FROM ((.DEVICE_CHAR[DVI_MAXBLOCK] + .CLUSTER_FACTOR[.RVN-1] - 1) /
;  2477          2473  3          .CLUSTER_FACTOR[.RVN-1])
;  2478          2474  2      TO .SMAP_SIZE[.RVN-1] * 512 * 8 - 1
;  2479          2475  2  DO
;  2480          2476  2      BITVECTOR[.NSMAP[.RVN-1], .N] = FALSE;
;  2481          2477  2
;  2482          2478  2
;  2483          2479  2  ! Allocate memory for and initialize allocated cluster bitmap.
;  2484          2480  2  !
;  2485          2481  2  STATUS = LIB$GET_VM(%REF(.SMAP_SIZE[.RVN-1] * 512), VSMAP[.RVN-1]);
;  2486          2482  2  IF NOT .STATUS THEN SIGNAL(VERIFY$_ALLOCMEM, 0, .STATUS);
;  2487          2483  2  MOVE(.SMAP_SIZE[.RVN-1] * 128, .NSMAP[.RVN-1], .VSMAP[.RVN-1]);
;  2488          2484  2
;  2489          2485  2
;  2490          2486  2  ! Allocate memory for and initialize multiply allocated cluster bitmap.
;  2491          2487  2  !
;  2492          2488  2  STATUS = LIB$GET_VM(%REF(.SMAP_SIZE[.RVN-1] * 512), MULTSMAP[.RVN-1]);
;  2493          2489  2  IF NOT .STATUS THEN SIGNAL(VERIFY$_ALLOCMEM, 0, .STATUS);
;  2494          2490  2  INITIALIZE(0, .SMAP_SIZE[.RVN-1] * 128, .MULTSMAP[.RVN-1]);
;  2495          2491  2
;  2496          2492  2
;  2497          2493  2  ! Delete the bitmap file window.
```

```
;  2498          2494  2 !
;  2499          2495  2 DELETE_WINDOW(.WINDOW);
;  2500          2496  2
;  2501          2497  2
;  2502          2498  2 ! Deaccess the storage bitmap file.
;  2503          2499  2 !
;  2504        P 2500  2 $QIOW(
;  2505        P 2501  2     FUNC=IO$_DEACCESS,
;  2506          2502  2     CHAN=.CHANNEL);
;  2507          2503  1 END;


                                                017D6                   .BLKB   2
                                          001A  0004  017D8 P.ABJ:      .WORD   4, 26                                   ;
                                            00000000' 017DC             .ADDRESS DEVICE_CHAR                            ;
                                            00000000  017E0             .LONG   0                                       ;
                                          0024  0004  017E4             .WORD   4, 36                                   ;
                                            00000000' 017E8             .ADDRESS DEVICE_CHAR+4                          ;
                                            00000000  017EC             .LONG   0                                       ;
                                          0026  0004  017F0             .WORD   4, 38                                   ;
                                            00000000' 017F4             .ADDRESS DEVICE_CHAR+8                          ;
                                            00000000  017F8             .LONG   0                                       ;
                                          0028  0004  017FC             .WORD   4, 40                                   ;
                                            00000000' 01800             .ADDRESS DEVICE_CHAR+12                         ;
                                            00000000  01804             .LONG   0                                       ;
                                          0034  0010  01808             .WORD   16, 52                                  ;
                                  00000000' 00000000' 0180C             .ADDRESS DEVICE_NAME, DEVICE_DESC               ;
                                            00000000  01814             .LONG   0                                       ;


                                                OFFC 00000 INIT_VOL_DATA:
                                                           .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11                 ; 1696
                                     5E             18  C2 00002       SUBL2   #24, SP                                 ;
                                  10 AE 00010001    8F  D0 00005       MOVL    #65537, INDEX_FILE_ID                   ; 1775
                                     57       04    AC  D0 0000D       MOVL    RVN, R7                                 ; 1777
                                  14 AE             57  B0 00011       MOVW    R7, INDEX_FILE_ID+4                     ;
                                  08 AE 00020002    8F  D0 00015       MOVL    #131074, BITMAP_FILE_ID                 ; 1778
                                  0C AE             57  B0 0001D       MOVW    R7, BITMAP_FILE_ID+4                    ; 1780
                10          00    6E             00  2C 00021       MOVC5   #0, (SP), #0, #T6, DEVICE_CHAR          ; 1785
                                            00000000' EF      00026
                                                       7E  7C 0002B       CLRQ    -(SP)                               ; 1799
                                                       7E  7C 0002D       CLRQ    -(SP)                               ;
                                                    8E AF  9F 0002F       PUSHAB  P.ABJ                               ;
                                                00000000' EF  9F 00032    PUSHAB  DEVICE_DESC                         ;
                                                       7E  7C 00038       CLRQ    -(SP)                               ;
                                     00000000G      00  08  FB 0003A     CALLS   #8, SYS$GETDVI                      ;
                                                    58  50  D0 00041     MOVL    R0, STATUS                          ;
                                                    12  58  E8 00044     BLBS    STATUS, 1$                          ; 1800
                                                    7E  57  7D 00047     MOVQ    R7, -(SP)                           ; 1802
                                                        01  DD 0004A     PUSHL   #1                                  ;
                                                00000000G  8F  DD 0004C     PUSHL   #VERIFY$_GETDVI                     ;
                                     00000000G      00  04  FB 00052     CALLS   #4, LIB$SIGNAL                      ;
                                                    54 00000000'FF47  DE 00059 1$:  MOVAL   @IMAP_SIZE[R7], R4      ; 1811
                                                    51 00000000'FF47  DE 00061     MOVAL   @MAXFILIDX[R7], R1       ; 1812
                                                    53 00000000'FF47  DE 00069     MOVAL   @CLUSTER_FACTOR[R7], R3  ; 1813
```

```
                                52 00000000'FF47   DE 00071          MOVAL   @BITMAP_OFFSET[R7], R2                ; 1814
                                50 00000000'FF47   DE 00079          MOVAL   @HEADER_OFFSET[R7], R0               ; 1815
                                02 00000000'   EF  D1 00081          CMPL    STRUCTURE_LEVEL, #2                  ; 1807
                                               2E  12 00088          BNEQ    2$                                   ;
                                56 00000000'   EF  3C 0008A          MOVZWL  BUFFER+14, CLUSTER                   ; 1810
                                55 00000000'   EF  3C 00091          MOVZWL  BUFFER+32, R5                        ; 1811
                          FC    A4             55  D0 00098          MOVL    R5, -4(R4)                           ;
                    54          55             0C  78 0009C          ASHL    #12, R5, R4                          ; 1812
                          FC    A1       FF    A4  9E 000A0          MOVAB   -1(R4), -4(R1)                       ;
                          FC    A3             56  D0 000A5          MOVL    CLUSTER, -4(R3)                      ; 1813
                FC  A2                         02  78 000A9          ASHL    #2, CLUSTER, -4(R2)                  ; 1814
                                FC    A2       D6 000AE              INCL    -4(R2)                               ;
                          FC    A0         6546  DE 000B1          MOVAL   (R5)[CLUSTER], -4(R0)                ; 1815
                                               24  11 000B6          BRB     3$                                   ; 1807
                                56             01  D0 000B8 2$:      MOVL    #1, CLUSTER                          ; 1819
                                55 00000000'   EF  3C 000BB          MOVZWL  BUFFER, R5                           ; 1820
                          FC    A4             55  D0 000C2          MOVL    R5, -4(R4)                           ;
                    54          55             0C  78 000C6          ASHL    #12, R5, R4                          ; 1821
                          FC    A1       FF    A4  9E 000CA          MOVAB   -1(R4), -4(R1)                       ;
                          FC    A3             01  D0 000CF          MOVL    #1, -4(R3)                           ; 1822
                          FC    A2             03  D0 000D3          MOVL    #3, -4(R2)                           ; 1823
                          FC    A0       02    A5  9E 000D7          MOVAB   2(R5), -4(R0)                        ; 1824
                                50 00000000'FF47   DE 000DC 3$:      MOVAL   @SMAP_SIZE[R7], R0                   ; 1826
                    51 00000000'   EF    FC    A3  C1 000E4          ADDL3   -4(R3), DEVICE_CHAR, R1              ; 1827
                                               51  D7 000ED          DECL    R1                                   ;
                                51          FC    A3  C6 000EF          DIVL2   -4(R3), R1                           ; 1828
                                51        0FFF    C1  9E 000F3          MOVAB   4095(R1), R1                         ; 1827
                FC  A0                  51 00001000    8F  C7 000F8    DIVL3   #4096, R1, -4(R0)                    ; 1828
                                               57  DD 00101          PUSHL   R7                                   ; 1833
                                   00000000'   EF  9F 00103          PUSHAB  HDR_BUFFER                           ;
                          0000V    CF             02  FB 00109          CALLS   #2, CREATE_WINDOW                    ;
                                5A                50  D0 0010E          MOVL    R0, WINDOW                           ;
                                7E                7E  7C 00111          CLRQ    -(SP)                                ; 1931
                                7E                01  7D 00113          MOVQ    #1, -(SP)                            ;
                                7E          0200  8F  3C 00116          MOVZWL  #512, -(SP)                          ;
                                   00000000'   EF  9F 0011B          PUSHAB  HDR_BUFFER_2                         ;
                                               7E  7C 00121          CLRQ    -(SP)                                ;
                                   00000000'   EF  9F 00123          PUSHAB  IOSB                                 ;
                                               31  DD 00129          PUSHL   #49                                  ;
                                   00000000'   EF  DD 0012B          PUSHL   CHANNEL                              ;
                                               7E  D4 00131          CLRL    -(SP)                                ;
                    00000000G   00             0C  FB 00133          CALLS   #12, SYS$QIOW                        ;
                                58             50  D0 0013A          MOVL    R0, STATUS                           ;
                                0A             58  E9 0013D          BLBC    STATUS, 4$                           ; 1932
                                58 00000000'   EF  3C 00140          MOVZWL  IOSB, STATUS                         ;
                                12             58  E8 00147          BLBS    STATUS, 5$                           ; 1933
                                7E             57  7D 0014A 4$:      MOVQ    R7, -(SP)                            ; 1935
                                               01  DD 0014D          PUSHL   #1                                   ;
                                   00000000G   8F  DD 0014F          PUSHL   #VERIFY$_READBOOT                    ;
                    00000000G   00             04  FB 00155          CALLS   #4, LIB$SIGNAL                       ;
                                02 00000000'   EF  D1 0015C 5$:      CMPL    STRUCTURE_LEVEL, #2                  ; 1940
                                               09  12 00163          BNEQ    6$                                   ;
                54              56             03  C5 00165          MULL3   #3, CLUSTER, R4                      ;
                                5B             54  D0 00169          MOVL    R4, R11                              ;
                                               03  11 0016C          BRB     7$                                   ;
                                5B             02  D0 0016E 6$:      MOVL    #2, R11                              ;
                                59             01  D0 00171 7$:      MOVL    #1, VBN                              ;
```

```
                                          01AD  31 00174 8$:    BRW     19$                                  ;
                        00000000'  EF       59  D1 00177 9$:    CMPL    VBN, HOMEVBN                         ; 1942
                                            F4  13 0017E        BEQL    8$                                   ;
                                            7E  7C 00180        CLRQ    -(SP)                                ; 1954
                                            7E  D4 00182        CLRL    -(SP)                                ;
                                            59  DD 00184        PUSHL   VBN                                  ;
                                  7E  0200  8F  3C 00186        MOVZWL  #512, -(SP)                          ;
                                 00000000'  EF  9F 0018B        PUSHAB  HDR_BUFFER_2                         ;
                                            7E  7C 00191        CLRQ    -(SP)                                ;
                                 00000000'  EF  9F 00193        PUSHAB  IOSB                                 ;
                                            31  DD 00199        PUSHL   #49                                  ;
                                 00000000'  EF  DD 0019B        PUSHL   CHANNEL                              ;
                                            7E  D4 001A1        CLRL    -(SP)                                ;
                        00000000G  00       0C  FB 001A3        CALLS   #12, SYS$QIOW                        ;
                                   58       50  D0 001AA        MOVL    R0, STATUS                           ;
                                   07       58  E9 001AD        BLBC    STATUS, 10$                          ; 1955
                                   58 00000000' EF  3C 001B0    MOVZWL  IOSB, STATUS                         ;
                                   03       58  E8 001B7 10$:   BLBS    STATUS, 11$                          ; 1965
                                          00AF  31 001BA        BRW     15$                                  ;
                                   02 00000000' EF  D1 001BD 11$: CMPL  STRUCTURE_LEVEL, #2                  ; 1968
                                            7B  12 001C4        BNEQ    12$                                  ;
                                            3A  DD 001C6        PUSHL   #58                                  ; 1973
                                 00000000'  EF  9F 001C8        PUSHAB  HDR_BUFFER_2                         ;
                        00000000G  EF       02  FB 001CE        CALLS   #2, CHECKSUM2                        ;
                                   7C       50  E9 001D5        BLBC    R0, 14$                              ;
                                  7E  01FE  8F  3C 001D8        MOVZWL  #510, -(SP)                          ; 1974
                                 00000000'  EF  9F 001DD        PUSHAB  HDR_BUFFER_2                         ;
                        00000000G  EF       02  FB 001E3        CALLS   #2, CHECKSUM2                        ;
                                   67       50  E9 001EA        BLBC    R0, 14$                              ;
                                        04  AE  9F 001ED        PUSHAB  TESTLBN                              ; 1975
                                            59  DD 001F0        PUSHL   VBN                                  ;
                                            5A  DD 001F2        PUSHL   WINDOW                               ;
                            0000V  CF       03  FB 001F4        CALLS   #3, MAP_VIRTUAL                      ;
                                   58       50  E9 001F9        BLBC    R0, 14$                              ;
                        00000000'  EF   04  AE  D1 001FC        CMPL    TESTLBN, HDR_BUFFER_2                ;
                                            4E  12 00204        BNEQ    14$                                  ;
        59 00000000'  EF           10       00  ED 00206        CMPZV   #0, #16, HDR_BUFFER_2+16, VBN        ; 1976
                                            43  12 0020F        BNEQ    14$                                  ;
                                 00000000'  EF  D4 00211        CLRL    BUFFER                               ; 1982
                                 00000000'  EF  B4 00217        CLRW    BUFFER+16                            ; 1983
                                 00000000'  EF  B4 0021D        CLRW    BUFFER+58                            ; 1984
                                 00000000'  EF  B4 00223        CLRW    BUFFER+510                           ; 1985
                                 00000000'  EF  D4 00229        CLRL    HDR_BUFFER_2                         ; 1986
                                 00000000'  EF  B4 0022F        CLRW    HDR_BUFFER_2+16                      ; 1987
                                 00000000'  EF  B4 00235        CLRW    HDR_BUFFER_2+58                      ; 1988
                                 00000000'  EF  B4 0023B        CLRW    HDR_BUFFER_2+510                     ; 1989
   00000000'  EF 00000000'  EF    0200      8F  29 00241 12$:   CMPC3   #512, HDR_BUFFER_2, BUFFER           ; 1994
                                            03  12 0024F        BNEQ    14$                                  ;
                                          00D0  31 00251 13$:   BRW     19$                                  ;
                                   15       58  E9 00254 14$:   BLBC    STATUS, 15$                          ; 2001
                                            57  DD 00257        PUSHL   R7                                   ; 2002
                                            59  DD 00259        PUSHL   VBN                                  ;
                                            02  DD 0025B        PUSHL   #2                                   ;
                                 00000000G  8F  DD 0025D        PUSHL   #VERIFY$_CHKALTHOME                  ;
                        00000000G  00       04  FB 00263        CALLS   #4, LIB$SIGNAL                       ;
                                            14  11 0026A        BRB     16$                                  ;
                                   7E       57  7D 0026C 15$:   MOVQ    R7, -(SP)                            ; 2003
```

```
                                                59  DD 0026F          PUSHL   VBN                                             ;
                                                02  DD 00271          PUSHL   #2                                              ;
                                 00000000G      8F  DD 00273          PUSHL   #VERIFY$_READHOME                               ;
                          00000000G   00        05  FB 00279          CALLS   #5, LIB$SIGNAL                                  ;
                              0000V   CF                   00  FB 00280 16$:  CALLS   #0, DO_REPAIR                          ; 2008
                                      C9                   50  E9 00285      BLBC    R0, 13$                                 ;
00000000'  EF 00000000'   EF      0200                     8F  28 00288      MOVC3   #512, BUFFER, HDR_BUFFER_2              ; 2011
                                      02 00000000'         EF  D1 00296      CMPL    STRUCTURE_LEVEL, #2                     ; 2012
                                                           37  12 0029D      BNEQ    17$                                     ;
                          00000000'   EF                   59  B0 0029F      MOVW    VBN, HDR_BUFFER_2+16                    ; 2015
                                      00000000'            EF  9F 002A6      PUSHAB  HDR_BUFFER_2                            ; 2016
                                                           59  DD 002AC      PUSHL   VBN                                     ;
                                                           5A  DD 002AE      PUSHL   WINDOW                                  ;
                              0000V   CF                   03  FB 002B0      CALLS   #3, MAP_VIRTUAL                         ;
                                                           3A  DD 002B5      PUSHL   #58                                     ; 2017
                                      00000000'            EF  9F 002B7      PUSHAB  HDR_BUFFER_2                            ;
                          00000000G   EF                   02  FB 002BD      CALLS   #2, CHECKSUM2                           ;
                                      7E       01FE        8F  3C 002C4      MOVZWL  #510, -(SP)                             ; 2018
                                      00000000'            EF  9F 002C9      PUSHAB  HDR_BUFFER_2                            ;
                          00000000G   EF                   02  FB 002CF      CALLS   #2, CHECKSUM2                           ;
                                                           7E  7C 002D6 17$:  CLRQ    -(SP)                                   ; 2026
                                                           7E  D4 002D8      CLRL    -(SP)                                   ;
                                                           59  DD 002DA      PUSHL   VBN                                     ;
                                      7E       0200        8F  3C 002DC      MOVZWL  #512, -(SP)                             ;
                                      00000000'            EF  9F 002E1      PUSHAB  HDR_BUFFER_2                            ;
                                                           7E  7C 002E7      CLRQ    -(SP)                                   ;
                                      00000000'            EF  9F 002E9      PUSHAB  IOSB                                    ;
                                                           30  DD 002EF      PUSHL   #48                                     ;
                                      00000000'            EF  DD 002F1      PUSHL   CHANNEL                                 ;
                                                           7E  D4 002F7      CLRL    -(SP)                                   ;
                          00000000G   00                   0C  FB 002F9      CALLS   #12, SYS$QIOW                           ;
                                      58                   50  D0 00300      MOVL    R0, STATUS                              ;
                                      0A                   58  E9 00303      BLBC    STATUS, 18$                             ; 2027
                                      58       00000000'   EF  3C 00306      MOVZWL  IOSB, STATUS                            ;
                                      14                   58  E8 0030D      BLBS    STATUS, 19$                             ; 2028
                                      7E                   57  7D 00310 18$:  MOVQ    R7, -(SP)                               ; 2030
                                                           59  DD 00313      PUSHL   VBN                                     ;
                                                           02  DD 00315      PUSHL   #2                                      ;
                                      00000000G            8F  DD 00317      PUSHL   #VERIFY$_WRITEHOME                      ;
                          00000000G   00                   05  FB 0031D      CALLS   #5, LIB$SIGNAL                          ;
FE4D                  59              01                   5B  F1 00324 19$:  ACBL    R11, #1, VBN, 9$                        ; 1940
                                      00000000'FF47        DF  0032A      PUSHAL  @IMAP[R7]                               ; 2039
                                      6E                   04  C2 00331      SUBL2   #4, (SP)                                ;
                                      50       00000000'FF47   DE 00334      MOVAL   @IMAP_SIZE[R7], R0                      ;
                  04  AE        FC   A0                   09  78 0033C      ASHL    #9, -4(R0), 4(SP)                       ;
                                                  04  AE   9F  00342      PUSHAB  4(SP)                                   ;
                          00000000G   00                   02  FB 00345      CALLS   #2, LIB$GET_VM                          ;
                                      58                   50  D0 0034C      MOVL    R0, STATUS                              ;
                                      11                   58  E8 0034F      BLBS    STATUS, 20$                             ; 2040
                                                           58  DD 00352      PUSHL   STATUS                                  ;
                                                           7E  D4 00354      CLRL    -(SP)                                   ;
                                      00000000G            8F  DD 00356      PUSHL   #VERIFY$_ALLOCMEM                       ;
                          00000000G   00                   03  FB 0035C      CALLS   #3, LIB$SIGNAL                          ;
                                      50       00000000'FF47   DE 00363 20$:  MOVAL   @IMAP_SIZE[R7], R0                      ; 2041
                  59          FC      A0                   01  C3 0036B      SUBL3   #1, -4(R0), R9                          ;
                                      52           81      8F  98 00370      CVTBL   #-127, VBN                              ;
                                                   00EB    31  00374 21$:  BRW     28$                                     ;
```

```
                        50 00000000'FF47   DE 00377 22$:    MOVAL   @IMAP_SIZE[R7], R0                      ; 2051
                    50      FC      A0     C3 0037F        SUBL3   VBN, -4(R0), R0
                0000007F    8F              D1 00384        CMPL    R0, #127                                ; 2049
                                        04  1B 0038B        BLEQU   23$
                        50      7F      8F  9A 0038D        MOVZBL  #127, R0
                        55              50  D0 00391 23$:   MOVL    R0, THIS_BLOCKS
                                        7E  7C 00394        CLRQ    -(SP)                                   ; 2063
                                        7E  D4 00396        CLRL    -(SP)
                        50 00000000'FF47   DE 00398        MOVAL   @BITMAP_OFFSET[R7], R0
                                FC B042    9F 003A0        PUSHAB  @-4(R0)[VBN]
                    7E          55      09  78 003A4        ASHL    #9, THIS_BLOCKS, -(SP)
                        50 00000000'FF47   DE 003A8        MOVAL   @IMAP[R7], R0
                    54          52      09  78 003B0        ASHL    #9, VBN, R4
                                FC B044    9F 003B4        PUSHAB  @-4(R0)[R4]
                                        7E  7C 003B8        CLRQ    -(SP)
                                00000000'  EF 9F 003BA      PUSHAB  IOSB
                                        31  DD 003C0        PUSHL   #49
                                00000000'  EF DD 003C2      PUSHL   CHANNEL
                                        7E  D4 003C8        CLRL    -(SP)
                00000000G       00          0C FB 003CA     CALLS   #12, SYS$QIOW
                                58          50 D0 003D1     MOVL    R0, STATUS
                                0A          58 E9 003D4     BLBC    STATUS, 24$                             ; 2064
                                58 00000000'EF 3C 003D7     MOVZWL  IOSB, STATUS
                                93          58 E8 003DE     BLBS    STATUS, 21$                             ; 2065
                                53          01 CE 003E1 24$:MNEGL   #1, XVBN                                ; 2068
                                            78 11 003E4     BRB     27$
                                        7E  7C 003E6 25$:   CLRQ    -(SP)                                   ; 2076
                                        7E  D4 003E8        CLRL    -(SP)
                        50 00000000'FF47   DE 003EA        MOVAL   @BITMAP_OFFSET[R7], R0
                    50          52      FC  A0 C1 003F2     ADDL3   -4(R0), VBN, R0
                                    6340    9F 003F7        PUSHAB  (XVBN)[R0]
                                7E    0200  8F 3C 003FA     MOVZWL  #512, -(SP)
                                50 00000000'FF47   DE 003FF        MOVAL   @IMAP[R7], R0
                    50          54      FC  A0 C1 00407     ADDL3   -4(R0), R4, R0
                    51          53          09 78 0040C     ASHL    #9, XVBN, R1
                                    6140    9F 00410        PUSHAB  (R1)[R0]
                                        7E  7C 00413        CLRQ    -(SP)
                                00000000'  EF 9F 00415      PUSHAB  IOSB
                                        31  DD 0041B        PUSHL   #49
                                00000000'  EF DD 0041D      PUSHL   CHANNEL
                                        7E  D4 00423        CLRL    -(SP)
                00000000G       00          0C FB 00425     CALLS   #12, SYS$QIOW
                                58          50 D0 0042C     MOVL    R0, STATUS
                                0A          58 E9 0042F     BLBC    STATUS, 26$                             ; 2077
                                58 00000000'EF 3C 00432     MOVZWL  IOSB, STATUS
                                22          58 E8 00439     BLBS    STATUS, 27$                             ; 2078
                                7E          57 7D 0043C 26$:MOVQ    R7, -(SP)                               ; 2084
                                50 00000000'FF47   DE 0043F        MOVAL   @BITMAP_OFFSET[R7], R0          ; 2083
                    50          52      FC  A0 C1 00447     ADDL3   -4(R0), VBN, R0
                                    6340    9F 0044C        PUSHAB  (XVBN)[R0]
                                        02  DD 0044F        PUSHL   #2                                      ; 2080
                                00000000G   8F DD 00451     PUSHL   #VERIFY$_READIBMAP
                00000000G       00          05 FB 00457     CALLS   #5, LIB$SIGNAL
                    84          53          55 F2 0045E 27$:AOBLSS  THIS_BLOCKS, XVBN, 25$                  ; 2068
FF0B                52 0000007F 8F          59 F1 00462 28$:ACBL    R9, #127, VBN, 22$                      ; 2041
                                50 00000000'FF47   DE 0046C        MOVAL   @EOF[R7], R0                    ; 2093
                                        FC  A0 D4 00474     CLRL    -4(R0)
```

```
                                02 00000000'   EF   D1 00477          CMPL    STRUCTURE_LEVEL, #2                          ; 2094
                                               0D   12 0047E          BNEQ    29$
                51 00000000'    EF             10   9C 00480          ROTL    #16, HDR_BUFFER+28, R1                       ; 2096
                          FC    A0       FF    A1   9E 00488          MOVAB   -1(R1), -4(R0)
                                53 00000000'FF47   DE 0048D 29$:      MOVAL   @IMAP_SIZE[R7], R3                           ; 2103
                53        FC    A3             07   78 00495          ASHL    #7, -4(R3), R3
                52                               53   D0 0049A          MOVL    R3, J
                                                 4C   11 0049D          BRB     32$
                50 00000000'FF47                 DE   0049F 30$:      MOVAL   @IMAP[R7], R0                                ; 2105
                50              FC    B042       D0   004A7          MOVL    @-4(R0)[J], R0
                                                 3D   13 004AC          BEQL    32$
                54 00000000'FF47                 DE   004AE          MOVAL   @EOF[R7], R4                                 ; 2108
                53              52               05   78 004B6          ASHL    #5, J, R3                                    ; 2109
                                                 50   DD 004BA          PUSHL   R0                                           ; 2110
                   00000000G    EF               01   FB 004BC          CALLS   #1, LEFT_ONE
                51              53               50   C1 004C3          ADDL3   R0, R3, R1                                   ; 2109
                50 00000000'FF47                 DE   004C7          MOVAL   @HEADER_OFFSET[R7], R0                       ; 2111
                51              FC    A0         C0   004CF          ADDL2   -4(R0), R1
                50 00000000'FF47                 DE   004D3          MOVAL   @EOF[R7], R0                                 ; 2112
                          FC    A0               51   D1 004DB          CMPL    R1, -4(R0)
                                                 04   1E 004DF          BGEQU   31$
                51        FC    A0               D0   004E1          MOVL    -4(R0), R1
                          FC    A4               51   D0 004E5 31$:     MOVL    R1, -4(R4)                                   ; 2108
                                                 03   11 004E9          BRB     33$                                          ; 2107
                B1                               52   F4 004EB 32$:     SOBGEQ  J, 30$                                       ; 2103
                                00000000'FF47    DF   004EE 33$:     PUSHAL  @DIRMAP[R7]                                  ; 2120
                6E                               04   C2 004F5          SUBL2   #4, (SP)
                50 00000000'FF47                 DE   004F8          MOVAL   @IMAP_SIZE[R7], R0
          04    AE        FC    A0               09   78 00500          ASHL    #9, -4(R0), 4(SP)
                                04    AE         9F   00506          PUSHAB  4(SP)
                   00000000G    00               02   FB 00509          CALLS   #2, LIB$GET_VM
                58                               50   D0 00510          MOVL    R0, STATUS
                11                               58   E8 00513          BLBS    STATUS, 34$                                  ; 2121
                                                 58   DD 00516          PUSHL   STATUS
                                                 7E   D4 00518          CLRL    -(SP)
                                00000000G        8F   DD 0051A          PUSHL   #VERIFY$_ALLOCMEM
                   00000000G    00               03   FB 00520          CALLS   #3, LIB$SIGNAL
                50 00000000'FF47                 DE   00527 34$:      MOVAL   @DIRMAP[R7], R0                              ; 2122
                52              FC    A0         D0   0052F          MOVL    -4(R0), P
                51 00000000'FF47                 DE   00533          MOVAL   @IMAP_SIZE[R7], R1
          50    FC        A1                     07   78 0053B          ASHL    #7, -4(R1), R0
                                                 02   11 00540          BRB     36$
                                                 82   D4 00542 35$:     CLRL    (P)+
                FB                               50   F4 00544 36$:     SOBGEQ  N, 35$
                                00000000'FF47    DF   00547          PUSHAL  @LOSTMAP[R7]                                 ; 2127
                6E                               04   C2 0054E          SUBL2   #4, (SP)
          04    AE        FC    A1               09   78 00551          ASHL    #9, -4(R1), 4(SP)
                                04    AE         9F   00557          PUSHAB  4(SP)
                   00000000G    00               02   FB 0055A          CALLS   #2, LIB$GET_VM
                58                               50   D0 00561          MOVL    R0, STATUS
                11                               58   E8 00564          BLBS    STATUS, 37$                                  ; 2128
                                                 58   DD 00567          PUSHL   STATUS
                                                 7E   D4 00569          CLRL    -(SP)
                                00000000G        8F   DD 0056B          PUSHL   #VERIFY$_ALLOCMEM
                   00000000G    00               03   FB 00571          CALLS   #3, LIB$SIGNAL
                50 00000000'FF47                 DE   00578 37$:      MOVAL   @LOSTMAP[R7], R0                             ; 2129
                52              FC    A0         D0   00580          MOVL    -4(R0), P
```

```
                                51 00000000'FF47   DE 00584          MOVAL   @IMAP_SIZE[R7], R1
                50        FC    A1                 07 78 0058C            ASHL    #7, -4(R1), R0
                                                   02 11 00591            BRB     39$
                                                   B2 D4 00593 38$:       CLRL    (P)+
                                FB                 50 F4 00595 39$:       SOBGEQ  N, 38$
                                      00000000'FF47   DF 00598            PUSHAL  @EXTMAP[R7]                     2134
                                6E                 04 C2 0059F            SUBL2   #4, (SP)
          04    AE        FC    A1                 09 78 005A2            ASHL    #9, -4(R1), 4(SP)
                                      04    AE     9F 005A8               PUSHAB  4(SP)
                     00000000G  00                 02 FB 005AB            CALLS   #2, LIB$GET_VM
                                58                 50 D0 005B2            MOVL    R0, STATUS
                                11                 58 E8 005B5            BLBS    STATUS, 40$                     2135
                                                   58 DD 005B8            PUSHL   STATUS
                                                   7E D4 005BA            CLRL    -(SP)
                                      00000000G    BF DD 005BC            PUSHL   #VERIFY$_ALLOCMEM
                     00000000G  00                 03 FB 005C2            CALLS   #3, LIB$SIGNAL
                                50 00000000'FF47   DE 005C9 40$:          MOVAL   @EXTMAP[R7], R0                 2136
                                52        FC       A0 D0 005D1            MOVL    -4(R0), P
                                51 00000000'FF47   DE 005D5               MOVAL   @IMAP_SIZE[R7], R1
                50        FC    A1                 07 78 005DD            ASHL    #7, -4(R1), R0
                                                   D2 11 005E2            BRB     42$
                                                   B2 D4 005E4 41$:       CLRL    (P)+
                                FB                 50 F4 005E6 42$:       SOBGEQ  N, 41$
                                      00000000'FF47   DF 005E9            PUSHAL  @SEQMAP[R7]                     2143
                                6E                 04 C2 005F0            SUBL2   #4, (SP)
          04    AE        FC    A1                 0D 78 005F3            ASHL    #13, -4(R1), 4(SP)
                                      04    AE     9F 005F9               PUSHAB  4(SP)
                     00000000G  00                 02 FB 005FC            CALLS   #2, LIB$GET_VM
                                58                 50 D0 00603            MOVL    R0, STATUS
                                11                 58 E8 00606            BLBS    STATUS, 43$                     2144
                                                   58 DD 00609            PUSHL   STATUS
                                                   7E D4 0060B            CLRL    -(SP)
                                      00000000G    BF DD 0060D            PUSHL   #VERIFY$_ALLOCMEM
                     00000000G  00                 03 FB 00613            CALLS   #3, LIB$SIGNAL
          03 00000000'  EF                         04 E0 0061A 43$:       BBS     #4, QUAL, 44$                   2147
                                                00AB 31 00622            BRW     47$
                                      00000000'FF47   DF 00625 44$:       PUSHAL  @OWNER[R7]                      2155
                                6E                 04 C2 0062C            SUBL2   #4, (SP)
                                50 00000000'FF47   DE 0062F               MOVAL   @IMAP_SIZE[R7], R0
          04    AE        FC    A0                 0E 78 00637            ASHL    #14, -4(R0), 4(SP)
                                      04    AE     9F 0063D               PUSHAB  4(SP)
                     00000000G  00                 02 FB 00640            CALLS   #2, LIB$GET_VM
                                58                 50 D0 00647            MOVL    R0, STATUS
                                11                 58 E8 0064A            BLBS    STATUS, 45$                     2156
                                                   58 DD 0064D            PUSHL   STATUS
                                                   7E D4 0064F            CLRL    -(SP)
                                      00000000G    8F DD 00651            PUSHL   #VERIFY$_ALLOCMEM
                     00000000G  0D                 03 FB 00657            CALLS   #3, LIB$SIGNAL
                                      00000000'FF47   DF 0065E 45$:       PUSHAL  @ALLOCATION[R7]                 2163
                                6E                 04 C2 00665            SUBL2   #4, (SP)
                                50 00000000'FF47   DE 00668               MOVAL   @IMAP_SIZE[R7], R0
          04    AE        FC    A0                 0E 78 00670            ASHL    #14, -4(R0), 4(SP)
                                      04    AE     9F 00676               PUSHAB  4(SP)
                     00000000G  00                 02 FB 00679            CALLS   #2, LIB$GET_VM
                                58                 50 D0 00680            MOVL    R0, STATUS
                                11                 58 E8 00683            BLBS    STATUS, 46$                     2164
                                                   58 DD 00686            PUSHL   STATUS
```

```
                                     7E  D4 00688        CLRL    -(SP)                            :
                          00000000G  8F  DD 0068A        PUSHL   #VERIFY$_ALLOCMEM                :
           00000000G  00             03  FB 00690        CALLS   #3, LIB$SIGNAL                   :
                          00000000'FF47  DF 00697 46$:   PUSHAL  @USAGE[R7]                       : 2171
                      6E             04  C2 0069E        SUBL2   #4, (SP)                         :
                      50  00000000'FF47  DE 006A1        MOVAL   @IMAP_SIZE[R7], R0               :
    04   AE      FC   A0             0E  78 006A9        ASHL    #14, -4(R0), 4(SP)               :
                                04   AE  9F 006AF        PUSHAB  4(SP)                            :
           00000000G  00             02  FB 006B2        CALLS   #2, LIB$GET_VM                   :
                      58             50  D0 006B9        MOVL    R0, STATUS                       :
                      11             58  E8 006BC        BLBS    STATUS, 47$                      : 2172
                                     58  DD 006BF        PUSHL   STATUS                           :
                                     7E  D4 006C1        CLRL    -(SP)                            :
                          00000000G  8F  DD 006C3        PUSHL   #VERIFY$_ALLOCMEM                :
           00000000G  00             03  FB 006C9        CALLS   #3, LIB$SIGNAL                   :
                      02  00000000'  EF  D1 006D0 47$:   CMPL    STRUCTURE_LEVEL, #2              : 2176
                                     03  13 006D7        BEQL    48$                              :
                                   0269  31 006D9        BRW     62$                              :
                          00000000'FF47  DF 006DC 48$:   PUSHAL  @BACKMAP[R7]                     : 2184
                      6E             04  C2 006E3        SUBL2   #4, (SP)                         :
                      50  00000000'FF47  DE 006E6        MOVAL   @IMAP_SIZE[R7], R0               :
    04   AE      FC   A0  00006000   8F  C5 006EE        MULL3   #24576, -4(R0), 4(SP)            :
                                04   AE  9F 006F8        PUSHAB  4(SP)                            :
           00000000G  00             02  FB 006FB        CALLS   #2, LIB$GET_VM                   :
                      58             50  D0 00702        MOVL    R0, STATUS                       :
                      11             58  E8 00705        BLBS    STATUS, 49$                      : 2185
                                     58  DD 00708        PUSHL   STATUS                           :
                                     7E  D4 0070A        CLRL    -(SP)                            :
                          00000000G  8F  DD 0070C        PUSHL   #VERIFY$_ALLOCMEM                :
           00000000G  00             03  FB 00712        CALLS   #3, LIB$SIGNAL                   :
                                     7E  7C 00719 49$:   CLRQ    -(SP)                            : 2197
                                     7E  D4 0071B        CLRL    -(SP)                            :
                      50  00000000'FF47  DE 0071D        MOVAL   @HEADER_OFFSET[R7], R0           :
         7E      FC   A0             01  C1 00725        ADDL3   #1, -4(R0), -(SP)                :
                      7E      0200   8F  3C 0072A        MOVZWL  #512, -(SP)                      :
                          00000000'  EF  9F 0072F        PUSHAB  HDR_BUFFER                       :
                                     7E  7C 00735        CLRQ    -(SP)                            :
                          00000000'  EF  9F 00737        PUSHAB  IOSB                             :
                                     31  DD 0073D        PUSHL   #49                              :
                          00000000'  EF  DD 0073F        PUSHL   CHANNEL                          :
                                     7E  D4 00745        CLRL    -(SP)                            :
           00000000G  00             0C  FB 00747        CALLS   #12, SYS$QIOW                    :
                      58             50  D0 0074E        MOVL    R0, STATUS                       :
                      0A             58  E9 00751        BLBC    STATUS, 50$                      : 2198
                      58  00000000'  EF  3C 00754        MOVZWL  IOSB, STATUS                     :
                      1C             58  E8 0075B        BLBS    STATUS, 51$                      : 2199
                                     58  DD 0075E 50$:   PUSHL   STATUS                           : 2202
                          00000000'  EF  9F 00760        PUSHAB  HDR_BUFFER                       :
                                18   AE  9F 00766        PUSHAB  INDEX_FILE_ID                    :
                          00000000G  8F  DD 00769        PUSHL   #VERIFY$_READHEADER              :
              0000V   CF             04  FB 0076F        CALLS   #4, HEADER_ERROR                 :
                          00000000'  EF  94 00774        CLRB    HDR_BUFFER+7                     : 2203
                                     7E  7C 0077A 51$:   CLRQ    -(SP)                            : 2216
                                     7E  D4 0077C        CLRL    -(SP)                            :
         54           56             03  C5 0077E        MULL3   #3, CLUSTER, R4                  :
                                01   A4  9F 00782        PUSHAB  1(R4)                            :
                      7E      0200   8F  3C 00785        MOVZWL  #512, -(SP)                      :
```

```
                                00000000'  EF  9F  0078A          PUSHAB  HDR_BUFFER_2                                    ;
                                           7E  7C  00790          CLRQ    -(SP)                                           ;
                                00000000'  EF  9F  00792          PUSHAB  IOSB                                            ;
                                           31  DD  00798          PUSHL   #49                                             ;
                                00000000'  EF  DD  0079A          PUSHL   CHANNEL                                         ;
                                           7E  D4  007A0          CLRL    -(SP)                                           ;
                    00000000G   00         0C  FB  007A2          CALLS   #12, SYS$QIOW                                   ;
                                58         50  D0  007A9          MOVL    R0, STATUS                                      ;
                                0A         58  E9  007AC          BLBC    STATUS, 52$                                     ; 2217
                                58 00000000'  EF  3C  007AF          MOVZWL  IOSB, STATUS                                    ;
                                1C            58  E8  007B6          BLBS    STATUS, 53$                                     ; 2218
                                              58  DD  007B9 52$:     PUSHL   STATUS                                          ; 2221
                                00000000'     EF  9F  007BB          PUSHAB  HDR_BUFFER_2                                    ;
                                        18    AE  9F  007C1          PUSHAB  INDEX_FILE_ID                                   ;
                                00000000G     8F  DD  007C4          PUSHL   #VERIFY$_READHEADER                             ;
                        0000V   CF            04  FB  007CA          CALLS   #4, HEADER_ERROR                                ;
                                00000000'     EF  94  007CF          CLRB    HDR_BUFFER_2+7                                  ; 2222
                                        10    AE  9F  007D5 53$:     PUSHAB  INDEX_FILE_ID                                   ; 2228
                                00000000'     EF  9F  007D8          PUSHAB  HDR_BUFFER                                      ;
                        0000V   CF            02  FB  007DE          CALLS   #2, VERIFY_HEADER                               ;
                                03            50  E8  007E3          BLBS    R0, 54$                                         ;
                                           00C4   31  007E6          BRW     58$                                             ;
                                        10    AE  9F  007E9 54$:     PUSHAB  INDEX_FILE_ID                                   ; 2233
                                00000000'     EF  9F  007EC          PUSHAB  HDR_BUFFER_2                                    ;
                        0000V   CF            02  FB  007F2          CALLS   #2, VERIFY_HEADER                               ;
                                55            50  E9  007F7          BLBC    R0, 56$                                         ;
        00000000'   EF  00000000'     EF  91  007FA          CMPB    HDR_BUFFER+58, HDR_BUFFER_2+58                          ; 2237
                                              48  12  00805          BNEQ    56$                                             ;
                                51 00000000'  EF  9A  00807          MOVZBL  HDR_BUFFER+1, R1                                ; 2239
                                53 00000000'  EF  9A  0080E          MOVZBL  HDR_BUFFER+2, R3                                ;
                                53            51  C2  00815          SUBL2   R1, R3                                          ;
                                50 00000000'  EF  9A  00818          MOVZBL  HDR_BUFFER_2+1, R0                              ; 2241
                                52 00000000'  EF  9A  0081F          MOVZBL  HDR_BUFFER_2+2, R2                              ;
                                52            50  C2  00826          SUBL2   R0, R2                                          ;
                                   00000000'EF40  3F  00829          PUSHAW  HDR_BUFFER_2[R0]                                ; 2238
                                   00000000'EF41  3F  00830          PUSHAW  HDR_BUFFER[R1]                                  ;
    52          00              9E            53  2D  00837          CMPC5   R3, @(SP)+, #0, R2, @(SP)+                      ;
                                              9E      0083C                                                                  ;
                                              10  12  0083D          BNEQ    56$                                             ;
                    00000000'   EF  00000000'     D1  0083F          CMPL    HDR_BUFFER+28, HDR_BUFFER_2+28                  ; 2243
                                              03  12  0084A          BNEQ    56$                                             ;
                                           00F6   31  0084C 55$:     BRW     62$                                             ;
                                              57  DD  0084F 56$:     PUSHL   R7                                              ; 2252
                                              01  DD  00851          PUSHL   #1                                              ;
                                00000000G     8F  DD  00853          PUSHL   #VERIFY$_ALTIHDBAD                              ;
                    00000000G   00            03  FB  00859          CALLS   #3, LIB$SIGNAL                                  ;
                        0000V   CF            00  FB  00860          CALLS   #0, DO_REPAIR                                   ; 2253
                                E4            50  E9  00865          BLBC    R0, 55$                                         ;
                                              7E  7C  00868          CLRQ    -(SP)                                           ; 2262
                                              7E  D4  0086A          CLRL    -(SP)                                           ;
                                        01    A4  9F  0086C          PUSHAB  1(R4)                                           ;
                                7E          0200  8F  3C  0086F          MOVZWL  #512, -(SP)                                     ;
                                00000000'     EF  9F  00874          PUSHAB  HDR_BUFFER                                      ;
                                              7E  7C  0087A          CLRQ    -(SP)                                           ;
                                00000000'     EF  9F  0087C          PUSHAB  IOSB                                            ;
                                              30  DD  00882          PUSHL   #48                                             ;
                                00000000'     EF  DD  00884          PUSHL   CHANNEL                                         ;
```

```
                                        7E  D4 0088A          CLRL    -(SP)
                    00000000G   00         0C  FB 0088C          CALLS   #12, SYS$QIOW
                                58         50  D0 00893          MOVL    R0, STATUS
                                0A         58  E9 00896          BLBC    STATUS, 57$                     ;  2263
                                58 00000000'   EF  3C 00899          MOVZWL  IOSB, STATUS
                                A9         58  E8 008A0          BLBS    STATUS, 55$                     ;  2264
                                           58  DD 008A3 57$:      PUSHL   STATUS                          ;  2268
                                00000000'  EF  9F 008A5          PUSHAB  HDR_BUFFER                      ;  2266
                                           77  11 008AB          BRB     60$                             ;  2274
                                      10   AE  9F 008AD 58$:      PUSHAB  INDEX_FILE_ID
                                00000000'  EF  9F 008B0          PUSHAB  HDR_BUFFER_2
                      0000V     CF         02  FB 008B6          CALLS   #2, VERIFY_HEADER
                                76         50  E9 008BB          BLBC    R0, 61$
                                           57  DD 008BE          PUSHL   R7                              ;  2281
                                           01  DD 008C0          PUSHL   #1
                                00000000G  8F  DD 008C2          PUSHL   #VERIFY$_PRIIHDBAD
                    00000000G   00         03  FB 008C8          CALLS   #3, LIB$SIGNAL
                      0000V     CF         00  FB 008CF          CALLS   #0, DO_REPAIR                   ;  2282
                                6E         50  E9 008D4          BLBC    R0, 62$                         ;  2291
                                           7E  7C 008D7          CLRQ    -(SP)
                                           7E  D4 008D9          CLRL    -(SP)
                                50 00000000'FF47  DE 008DB          MOVAL   @HEADER_OFFSET[R7], R0
                7E      FC      A0         01  C1 008E3          ADDL3   #1, -4(R0), -(SP)
                                7E      0200   8F  3C 008E8          MOVZWL  #512, -(SP)
                                00000000'  EF  9F 008EC          PUSHAB  HDR_BUFFER_2
                                           7E  7C 008F3          CLRQ    -(SP)
                                00000000'  EF  9F 008F5          PUSHAB  IOSB
                                           30  DD 008FB          PUSHL   #48
                                00000000'  EF  DD 008FD          PUSHL   CHANNEL
                                           7E  D4 00903          CLRL    -(SP)
                    00000000G   00         0C  FB 00905          CALLS   #12, SYS$QIOW
                                58         50  D0 0090C          MOVL    R0, STATUS
                                0A         58  E9 0090F          BLBC    STATUS, 59$                     ;  2292
                                58 00000000'   EF  3C 00912          MOVZWL  IOSB, STATUS
                                29         58  E8 00919          BLBS    STATUS, 62$                     ;  2293
                                           58  DD 0091C 59$:      PUSHL   STATUS                          ;  2297
                                00000000'  EF  9F 0091E          PUSHAB  HDR_BUFFER_2                    ;  2295
                                      18   AE  9F 00924 60$:      PUSHAB  INDEX_FILE_ID
                                00000000G  8F  DD 00927          PUSHL   #VERIFY$_WRITEHEADER
                      0000V     CF         04  FB 0092D          CALLS   #4, HEADER_ERROR
                                           11  11 00932          BRB     62$                             ;  2274
                                           57  DD 00934 61$:      PUSHL   R7                              ;  2301
                                           01  DD 00936          PUSHL   #1
                                00000000G  8F  DD 00938          PUSHL   #VERIFY$_FINDIHD
                    00000000G   00         03  FB 0093E          CALLS   #3, LIB$SIGNAL
                                           5A  DD 00945 62$:      PUSHL   WINDOW                          ;  2308
                      0000V     CF         01  FB 00947          CALLS   #1, DELETE_WINDOW               ;  2315
                                           7E  7C 0094C          CLRQ    -(SP)
                                           7E  7C 0094E          CLRQ    -(SP)
                                           7E  7C 00950          CLRQ    -(SP)
                                           7E  7C 00952          CLRQ    -(SP)
                                      7E   34  7D 00954          MOVQ    #52, -(SP)
                                00000000'  EF  DD 00957          PUSHL   CHANNEL
                                           7E  D4 0095D          CLRL    -(SP)
                    00000000G   00         0C  FB 0095F          CALLS   #12, SYS$QIOW
0040    8F          00         6E         00  2C 00966          MOVC5   #0, (SP), #0, #64, FIB          ;  2320
                                00000000'  EF      0096D
```

```
                00000000'  EF      08   AC  D0 00972          MOVL    ACCTL_VALUE, FIB                        ; 2321
                00000000'  EF 00020002   8F  D0 0097A          MOVL    #131074, FIB+4                          ; 2322
                00000000'  EF            57  B0 00985          MOVW    R7, FIB+8                               ; 2324
                                         7E  D4 0098C          CLRL    -(SP)                                   ; 2330
                                    DE7A CF  9F 0098E          PUSHAB  HDR_ATR_DESC
                                         7E  7C 00992          CLRQ    -(SP)
                                         7E  D4 00994          CLRL    -(SP)
                                    DE7E CF  9F 00996          PUSHAB  FIB_DESC
                                         7E  7C 0099A          CLRQ    -(SP)
                                00000000'  EF  9F 0099C          PUSHAB  IOSB
                                      7E   72  8F  9A 009A2          MOVZBL  #114, -(SP)
                                00000000'  EF  DD 009A6          PUSHL   CHANNEL
                                         7E  D4 009AC          CLRL    -(SP)
                00000000G  00            0C  FB 009AE          CALLS   #12, SYS$QIOW
                           58            50  D0 009B5          MOVL    R0, STATUS
                           0A            58  E9 009B8          BLBC    STATUS, 63$                             ; 2331
                           58 00000000'  EF  3C 009BB          MOVZWL  IOSB, STATUS
                           12            58  E8 009C2          BLBS    STATUS, 64$                             ; 2332
                           7E            57  7D 009C5 63$:     MOVQ    R7, -(SP)                               ; 2334
                                         01  DD 009C8          PUSHL   #1
                                00000000G  8F  DD 009CA          PUSHL   #VERIFY$_OPENBITMAP
                00000000G  00            04  FB 009D0          CALLS   #4, LIB$SIGNAL
                                         57  DD 009D7 64$:     PUSHL   R7                                      ; 2339
                                00000000'  EF  9F 009D9          PUSHAB  HDR_BUFFER
                     0000V  CF            02  FB 009DF          CALLS   #2, CREATE_WINDOW
                           5A            50  D0 009E4          MOVL    R0, WINDOW
                                         7E  7C 009E7          CLRQ    -(SP)                                   ; 2350
                                      7E 01  7D 009E9          MOVQ    #1, -(SP)
                                      7E 0200  8F  3C 009EC          MOVZWL  #512, -(SP)
                                00000000'  EF  9F 009F1          PUSHAB  BUFFER
                                         7E  7C 009F7          CLRQ    -(SP)
                                00000000'  EF  9F 009F9          PUSHAB  IOSB
                                         31  DD 009FF          PUSHL   #49
                                00000000'  EF  DD 00A01          PUSHL   CHANNEL
                                         7E  D4 00A07          CLRL    -(SP)
                00000000G  00            0C  FB 00A09          CALLS   #12, SYS$QIOW
                           58            50  D0 00A10          MOVL    R0, STATUS
                           0A            58  E9 00A13          BLBC    STATUS, 65$                             ; 2351
                           58 00000000'  EF  3C 00A16          MOVZWL  IOSB, STATUS
                           18            58  E8 00A1D          BLBS    STATUS, 66$                             ; 2352
                           7E            57  7D 00A20 65$:     MOVQ    R7, -(SP)                               ; 2354
                                         01  DD 00A23          PUSHL   #1
                                00000000G  8F  DD 00A25          PUSHL   #VERIFY$_READSCB
                00000000G  00            04  FB 00A2B          CALLS   #4, LIB$SIGNAL
                           03            58  E8 00A32          BLBS    STATUS, 66$                             ; 2361
                                   00D7  31  00A35          BRW     75$
                           02 00000000'  EF  D1 00A38 66$:     CMPL    STRUCTURE_LEVEL, #2                     ; 2364
                                         03  13 00A3F          BEQL    67$
                                   0080  31  00A41          BRW     70$
                                00000000'  EF  9F 00A44 67$:     PUSHAB  BUFFER                                  ; 2367
                00000000G  EF            01  FB 00A4A          CALLS   #1, CHECKSUM
                           03            50  E8 00A51          BLBS    R0, 69$
                                   00A4  31  00A54 68$:     BRW     74$
                     0201  8F 00000000'  EF  B1 00A57 69$:     CMPW    BUFFER, #513                            ; 2368
                                         F2  12 00A60          BNEQ    68$
          56 00000000'  EF          10   00  ED 00A62          CMPZV   #0, #16, BUFFER+2, CLUSTER              ; 2369
                                         E7  12 00A6B          BNEQ    68$
```

```
                    00000000'  EF  00000000'  EF  D1  00A6D          CMPL    BUFFER+4, DEVICE_CHAR                                  ; 2370
                                                DA  12  00A78          BNEQ    68$
                 50 00000000'  EF  00000000'  EF  C5  00A7A          MULL3   DEVICE_CHAR+8, DEVICE_CHAR+4, R0                ; 2372
                               50  00000000'  EF  C4  00A86          MULL2   DEVICE_CHAR+12, R0                             ; 2373
                               50  00000000'  EF  C6  00A8D          DIVL2   DEVICE_CHAR, R0
                               50  00000000'  EF  D1  00A94          CMPL    BUFFER+8, R0
                                               5E  12  00A9B          BNEQ    74$
                    00000000'  EF  00000000'  EF  D1  00A9D          CMPL    BUFFER+12, DEVICE_CHAR+4                       ; 2374
                                               51  12  00AA8          BNEQ    74$
                    00000000'  EF  00000000'  EF  D1  00AAA          CMPL    BUFFER+16, DEVICE_CHAR+8                       ; 2375
                                               44  12  00AB5          BNEQ    74$
                    00000000'  EF  00000000'  EF  D1  00AB7          CMPL    BUFFER+20, DEVICE_CHAR+12                      ; 2376
                                               32  11  00AC2          BRB     72$
                               50  00000000'FF47  DE  00AC4  70$:     MOVAL   @SMAP_SIZE[R7], R0                             ; 2385
                               50        FC    A0  D0  00ACC          MOVL    -4(R0), BLOCK_COUNT
                    0000007E   8F              50  D1  00AD0          CMPL    BLOCK_COUNT, #126                              ; 2386
                                               02  1B  00AD7          BLEQU   71$
                                               50  D4  00AD9          CLRL    BLOCK_COUNT
 50 00000000'  EF              08               00  ED  00ADB  71$:     CMPZV   #0, #8, BUFFER+3, BLOCK_COUNT                  ; 2387
                                               15  12  00AE4          BNEQ    74$
                 51 00000000'  EF               10  9C  00AE6          ROTL    #16, DEVICE_CHAR, R1                           ; 2388
                               51  00000000'EF40  D1  00AEE          CMPL    BUFFER+4[BLOCK_COUNT], R1
                                               03  12  00AF6  72$:     BNEQ    74$
                                           011B  31  00AF8  73$:     BRW     82$
                               11              58  E9  00AFB  74$:     BLBC    STATUS, 75$                                    ; 2393
                                               57  DD  00AFE          PUSHL   R7
                                               01  DD  00B00          PUSHL   #1
                                   00000000G   8F  DD  00B02          PUSHL   #VERIFY$_CHKSCB
                    00000000G  00              03  FB  00B08          CALLS   #3, LIB$SIGNAL
                        0000V  CF              00  FB  00B0F  75$:     CALLS   #0, DO_REPAIR                                  ; 2394
                               E1              50  E9  00B14          BLBC    R0, 73$
0200   8F              00      6E              00  2C  00B17          MOVC5   #0, (SP), #0, #512, BUFFER                     ; 2400
                                   00000000'   EF      00B1E
                               52  00000000'   EF  D0  00B23          MOVL    DEVICE_CHAR, R2                                ; 2406
                               02  00000000'   EF  D1  00B2A          CMPL    STRUCTURE_LEVEL, #2                            ; 2401
                                               5E  12  00B31          BNEQ    76$
                    00000000'  EF       0201   8F  B0  00B33          MOVW    #513, BUFFER                                   ; 2404
                    00000000'  EF              56  B0  00B3C          MOVW    CLUSTER, BUFFER+2                              ; 2405
                    00000000'  EF              52  D0  00B43          MOVL    R2, BUFFER+4                                   ; 2406
                 50 00000000'  EF  00000000'  EF  C5  00B4A          MULL3   DEVICE_CHAR+8, DEVICE_CHAR+4, R0               ; 2408
                               50  00000000'  EF  C4  00B56          MULL2   DEVICE_CHAR+12, R0                             ; 2409
          00000000'  EF        50              52  C7  00B5D          DIVL3   R2, R0, BUFFER+8
                    00000000'  EF  00000000'  EF  7D  00B65          MOVQ    DEVICE_CHAR+4, BUFFER+12                       ; 2410
                    00000000'  EF  00000000'  EF  D0  00B70          MOVL    DEVICE_CHAR+12, BUFFER+20                      ; 2412
                    00000000'  EF              16  D0  00B7B          MOVL    #22, BUFFER+24                                 ; 2414
                                   00000000'   EF  9F  00B82          PUSHAB  BUFFER                                         ; 2415
                    00000000G  EF              01  FB  00B88          CALLS   #1, CHECKSUM
                                               3A  11  00B8F          BRB     80$                                            ; 2401
                               50  00000000'FF47  DE  00B91  76$:     MOVAL   @SMAP_SIZE[R7], R0                             ; 2424
                               50        FC    A0  D0  00B99          MOVL    -4(R0), BLOCK_COUNT
                    0000007E   8F              50  D1  00B9D          CMPL    BLOCK_COUNT, #126                              ; 2425
                                               02  1B  00BA4          BLEQU   77$
                                               50  D4  00BA6          CLRL    BLOCK_COUNT
                    00000000'  EF              50  90  00BA8  77$:     MOVB    BLOCK_COUNT, BUFFER+3                          ; 2426
                               51              01  CE  00BAF          MNEGL   #1, J                                          ; 2427
                                               0A  11  00BB2          BRB     79$
                    00000000'EF41      1000    8F  3C  00BB4  78$:     MOVZWL  #4096, BUFFER+4[J]
```

```
                F2            51            50  F2 00BBE 79$:  AOBLSS  BLOCK_COUNT, J, 78$
     00000000'EF40            52            10  9C 00BC2       ROTL    #16, R2, BUFFER+4[BLOCK_COUNT]        ; 2428
                                            7E  7C 00BCB 80$:  CLRQ    -(SP)                                 ; 2440
                              7E            01  7D 00BCD       MOVQ    #1, -(SP)
                              7E      0200  8F  3C 00BD0       MOVZWL  #512, -(SP)
                               00000000'    EF  9F 00BD5       PUSHAB  BUFFER
                                            7E  7C 00BDB       CLRQ    -(SP)
                               00000000'    EF  9F 00BDD       PUSHAB  IOSB
                                            30  DD 00BE3       PUSHL   #48
                               00000000'    EF  DD 00BE5       PUSHL   CHANNEL
                                            7E  D4 00BEB       CLRL    -(SP)
                    00000000G 00            0C  FB 00BED       CALLS   #12, SYS$QIOW
                              58            50  D0 00BF4       MOVL    R0, STATUS
                              0A            58  E9 00BF7       BLBC    STATUS, 81$                           ; 2441
                              58 00000000'  EF  3C 00BFA       MOVZWL  IOSB, STATUS
                              12            58  E8 00C01       BLBS    STATUS, 82$                           ; 2442
                              7E            57  7D 00C04 81$:  MOVQ    R7, -(SP)                             ; 2444
                                            01  DD 00C07       PUSHL   #1
                               00000000G    8F  DD 00C09       PUSHL   #VERIFY$_WRITESCB
                    00000000G 00            04  FB 00C0F       CALLS   #4, LIB$SIGNAL
                                            5A  D5 00C16 82$:  TSTL    WINDOW                                ; 2453
                                            19  13 00C18       BEQL    83$
                              01        04  AA  D1 00C1A       CMPL    4(WINDOW), #1                         ; 2457
                                            13  12 00C1E       BNEQ    83$
                              50 00000000'FF47  DE 00C20       MOVAL   @SMAP_SIZE[R7], R0                    ; 2458
                50        FC  A0            01  C1 00C28       ADDL3   #1, -4(R0), R0
                              50        08  AA  D1 00C2D       CMPL    8(WINDOW), R0
                                            11  1E 00C31       BGEQU   84$
                                            57  DD 00C33 83$:  PUSHL   R7                                    ; 2461
                                            01  DD 00C35       PUSHL   #1
                               00000000G    8F  DD 00C37       PUSHL   #VERIFY$_BADBITMAP
                    00000000G 00            03  FB 00C3D       CALLS   #3, LIB$SIGNAL
                               00000000'FF47    DF 00C44 84$:  PUSHAL  @NSMAP[R7]                            ; 2468
                              6E            04  C2 00C4B       SUBL2   #4, (SP)
                              50 00000000'FF47  DE 00C4E       MOVAL   @SMAP_SIZE[R7], R0
          04    AE        FC  A0            09  78 00C56       ASHL    #9, -4(R0), 4(SP)
                                        04  AE  9F 00C5C       PUSHAB  4(SP)
                    00000000G 00            02  FB 00C5F       CALLS   #2, LIB$GET_VM
                              58            50  D0 00C66       MOVL    R0, STATUS
                              11            58  E8 00C69       BLBS    STATUS, 85$                           ; 2469
                                            58  DD 00C6C       PUSHL   STATUS
                                            7E  D4 00C6E       CLRL    -(SP)
                               00000000G    8F  DD 00C70       PUSHL   #VERIFY$_ALLOCMEM
                    00000000G 00            03  FB 00C76       CALLS   #3, LIB$SIGNAL
                              52 00000000'FF47  DE 00C7D 85$:  MOVAL   @NSMAP[R7], R2                        ; 2470
                              53        FC  A2  D0 00C85       MOVL    -4(R2), P
                              51 00000000'FF47  DE 00C89       MOVAL   @SMAP_SIZE[R7], R1
                50        FC  A1            07  78 00C91       ASHL    #7, -4(R1), R0
                                            03  11 00C96       BRB     87$
                              83            01  CE 00C98 86$:  MNEGL   #1, (P)+
                              FA            50  F4 00C9B 87$:  SOBGEQ  N, 86$
                              50 00000000'FF47  DE 00C9E       MOVAL   @CLUSTER_FACTOR[R7], R0               ; 2472
                53 00000000'  EF        FC  A0  C1 00CA6       ADDL3   -4(R0), DEVICE_CHAR, R3
                                            53  D7 00CAF       DECL    R3
                50            53        FC  A0  C7 00CB1       DIVL3   -4(R0), R3, R0                        ; 2473
                53        FC  A1            0C  78 00CB6       ASHL    #12, -4(R1), R3                       ; 2474
                                            50  D7 00CBB       DECL    N                                     ; 2471
```

```
                                 05  11 00CBD        BRB     89$
          00        FC  B2       50  E5 00CBF 88$:   BBCC    N, @-4(R2), 89$                    ; 2476
          F7            50       53  F2 00CC4 89$:   AOBLSS  R3, N, 88$
                   00000000'FF47 DF 00CC8            PUSHAL  @VSMAP[R7]                         ; 2481
                   6E            04  C2 00CCF        SUBL2   #4, (SP)
     04   AE        FC  A1       09  78 00CD2        ASHL    #9, -4(R1), 4(SP)
                            04   AE  9F 00CD8        PUSHAB  4(SP)
          00000000G  00          02  FB 00CDB        CALLS   #2, LIB$GET_VM
                     58          50  D0 00CE2        MOVL    R0, STATUS
                     11          58  E8 00CE5        BLBS    STATUS, 90$                        ; 2482
                                 58  DD 00CE8        PUSHL   STATUS
                                 7E  D4 00CEA        CLRL    -(SP)
                   00000000G     8F  DD 00CEC        PUSHL   #VERIFY$_ALLOCMEM
          00000000G  00          03  FB 00CF2        CALLS   #3, LIB$SIGNAL
                     50 00000000'FF47 DE 00CF9 90$:  MOVAL   @NSMAP[R7], R0                     ; 2483
                     53      FC  A0  D0 00D01        MOVL    -4(R0), P
                     50 00000000'FF47 DE 00D05       MOVAL   @VSMAP[R7], R0
                     52      FC  A0  D0 00D0D        MOVL    -4(R0), Q
                     51 00000000'FF47 DE 00D11       MOVAL   @SMAP_SIZE[R7], R1
          50        FC  A1       07  78 00D19        ASHL    #7, -4(R1), R0
                                 03  11 00D1E        BRB     92$
                     82          83  D0 00D20 91$:   MOVL    (P)+, (Q)+
                     FA          50  F4 00D23 92$:   SOBGEQ  N, 91$
                   00000000'FF47 DF 00D26            PUSHAL  @MULTSMAP[R7]                      ; 2488
                   6E            04  C2 00D2D        SUBL2   #4, (SP)
     04   AE        FC  A1       09  78 00D30        ASHL    #9, -4(R1), 4(SP)
                            04   AE  9F 00D36        PUSHAB  4(SP)
          00000000G  00          02  FB 00D39        CALLS   #2, LIB$GET_VM
                     58          50  D0 00D40        MOVL    R0, STATUS
                     11          58  E8 00D43        BLBS    STATUS, 93$                        ; 2489
                                 58  DD 00D46        PUSHL   STATUS
                                 7E  D4 00D48        CLRL    -(SP)
                   00000000G     8F  DD 00D4A        PUSHL   #VERIFY$_ALLOCMEM
          00000000G  00          03  FB 00D50        CALLS   #3, LIB$SIGNAL
                     50 00000000'FF47 DE 00D57 93$:  MOVAL   @MULTSMAP[R7], R0                  ; 2490
                     51      FC  A0  D0 00D5F        MOVL    -4(R0), P
                     50 00000000'FF47 DE 00D63       MOVAL   @SMAP_SIZE[R7], R0
          50        FC  A0       07  78 00D6B        ASHL    #7, -4(R0), R0
                                 02  11 00D70        BRB     95$
                                 81  D4 00D72 94$:   CLRL    (P)+
                     FB          50  F4 00D74 95$:   SOBGEQ  N, 94$
                                 5A  DD 00D77        PUSHL   WINDOW                             ; 2495
               0000V CF          01  FB 00D79        CALLS   #1, DELETE_WINDOW
                                 7E  7C 00D7E        CLRQ    -(SP)                              ; 2502
                                 7E  7C 00D80        CLRQ    -(SP)
                                 7E  7C 00D82        CLRQ    -(SP)
                                 7E  7C 00D84        CLRQ    -(SP)
                     7E          34  7D 00D86        MOVQ    #52, -(SP)
                   00000000'     EF  DD 00D89        PUSHL   CHANNEL
                                 7E  D4 00D8F        CLRL    -(SP)
          00000000G  00          0C  FB 00D91        CALLS   #12, SYS$QIOW
                                     04 00D98        RET                                        ; 2503

; Routine Size:  3481 bytes,    Routine Base:  CODE + 1818
```

```
;  2509     2504  1 ROUTINE READ_HOMEBLOCK(RVN): NOVALUE=
;  2510     2505  1
;  2511     2506  1 !++
;  2512     2507  1 !
;  2513     2508  1 ! FUNCTIONAL DESCRIPTION:
;  2514     2509  1 !       This routine reads the first good home block of the currently open
;  2515     2510  1 !       index file into the general buffer.  The code in this routine should
;  2516     2511  1 !       track the code in MOUNT.
;  2517     2512  1 !
;  2518     2513  1 ! INPUT PARAMETERS:
;  2519     2514  1 !       RVN             - Relative volume number.
;  2520     2515  1 !
;  2521     2516  1 ! IMPLICIT INPUTS:
;  2522     2517  1 !       NONE
;  2523     2518  1 !
;  2524     2519  1 ! OUTPUT PARAMETERS:
;  2525     2520  1 !       NONE
;  2526     2521  1 !
;  2527     2522  1 ! IMPLICIT OUTPUTS:
;  2528     2523  1 !       BUFFER          - Contains a valid home block.
;  2529     2524  1 !       VOLUME_COUNT    - Count of volumes in volume set.
;  2530     2525  1 !       STRUCTURE_LEVEL - Structure level (1 or 2) of the volume set.
;  2531     2526  1 !       HOMEVBN         - VBN of the valid home block.
;  2532     2527  1 !
;  2533     2528  1 ! ROUTINE VALUE:
;  2534     2529  1 !       NONE
;  2535     2530  1 !
;  2536     2531  1 ! SIDE EFFECTS:
;  2537     2532  1 !       NONE
;  2538     2533  1 !
;  2539     2534  1 !--
;  2540     2535  1
;  2541     2536  2 BEGIN
;  2542     2537  2 LOCAL
;  2543     2538  2         STATUS,                         ! General status value
;  2544     2539  2         OLD_STATUS;                     ! Save status for error message
;  2545     2540  2
;  2546     2541  2
;  2547     2542  2 ! We keep reading until we get a block that reads without errors and looks
;  2548     2543  2 ! like a home block.  Track any error status for the eventual error message.
;  2549     2544  2 !
;  2550     2545  2 OLD_STATUS = SS$_ABORT;
;  2551     2546  2 INCR VBN FROM 2 TO 100 DO
;  2552     2547  3     BEGIN
;  2553   P 2548  3     STATUS = $QIOW(
;  2554   P 2549  3         FUNC=IO$_READVBLK,
;  2555   P 2550  3         CHAN=.CHANNEL,
;  2556   P 2551  3         IOSB=IOSB,
;  2557   P 2552  3         P1=BUFFER,
;  2558   P 2553  3         P2=512,
;  2559     2554  3         P3=.VBN);
;  2560     2555  3     IF .STATUS THEN STATUS = .IOSB[0];
;  2561     2556  3
;  2562     2557  3     IF NOT .STATUS
;  2563     2558  3     THEN
;  2564     2559  3         OLD_STATUS = .STATUS
;  2565     2560  3     ELSE
```

```
;  2566     2561  3          IF
;  2567     2562  3              .BUFFER[HM2$B_STRUCLEV] EQL 2 AND
;  2568     2563  3              .BUFFER[HM2$L_ALTIDXLBN] NEQ 0 AND
;  2569     2564  3              .BUFFER[HM2$W_CLUSTER] NEQ 0 AND
;  2570     2565  3              .BUFFER[HM2$W_HOMEVBN] NEQ 0 AND
;  2571     2566  3              .BUFFER[HM2$W_ALHOMEVBN] NEQ 0 AND
;  2572     2567  3              .BUFFER[HM2$W_ALTIDXVBN] NEQ 0 AND
;  2573     2568  3              .BUFFER[HM2$W_IBMAPVBN] NEQ 0 AND
;  2574     2569  3              .BUFFER[HM2$L_IBMAPLBN] NEQ 0 AND
;  2575     2570  3              .BUFFER[HM2$L_MAXFILES] NEQ 0 AND
;  2576     2571  3              .BUFFER[HM2$W_IBMAPSIZE] NEQ 0 AND
;  2577     2572  3              .BUFFER[HM2$W_RESFILES] NEQ 0 AND
;  2578     2573  3              CHECKSUM2(BUFFER, $BYTEOFFSET(HM2$W_CHECKSUM1)) AND
;  2579     2574  3              CHECKSUM2(BUFFER, $BYTEOFFSET(HM2$W_CHECKSUM2))
;  2580     2575  3          THEN
;  2581     2576  4              BEGIN
;  2582     2577  4
;  2583     2578  4              ! Block is a valid ODS-2 home block.
;  2584     2579  4              !
;  2585     2580  4              IF .RVN EQL 1
;  2586     2581  4              THEN
;  2587     2582  5                  BEGIN
;  2588     2583  5                  STRUCTURE_LEVEL = 2;
;  2589     2584  5                  VOLUME_COUNT = .BUFFER[HM2$W_SETCOUNT];
;  2590     2585  5                  IF .VOLUME_COUNT EQL 0 THEN VOLUME_COUNT = 1;
;  2591     2586  5                  IF .VOLUME_COUNT GTRU MAX_VOLUMES THEN SIGNAL(VERIFY$_MAXVOLS);
;  2592     2587  4                  END;
;  2593     2588  4              HOMEVBN = .VBN;
;  2594     2589  4              RETURN;
;  2595     2590  4              END
;  2596     2591  3          ELSE IF
;  2597     2592  3              .RVN EQL 1 AND
;  2598     2593  3              (.BUFFER[HM1$W_STRUCLEV] EQL HM1$C_LEVEL1 OR .BUFFER[HM1$W_STRUCLEV] EQL HM1$C_LEVEL2) AND
;  2599     2594  3              .BUFFER[HM1$W_CLUSTER] NEQ 0 AND
;  2600     2595  3              .BUFFER[HM1$L_IBMAPLBN] NEQ 0 AND
;  2601     2596  3              .BUFFER[HM1$W_MAXFILES] NEQ 0 AND
;  2602     2597  3              .BUFFER[HM1$W_IBMAPSIZE] NEQ 0 AND
;  2603     2598  3              CHECKSUM2(BUFFER, $BYTEOFFSET(HM1$W_CHECKSUM1)) AND
;  2604     2599  3              CHECKSUM2(BUFFER, $BYTEOFFSET(HM1$W_CHECKSUM2))
;  2605     2600  3          THEN
;  2606     2601  4              BEGIN
;  2607     2602  4
;  2608     2603  4              ! Block is a valid ODS-1 home block.
;  2609     2604  4              !
;  2610     2605  4              STRUCTURE_LEVEL = 1;
;  2611     2606  4              VOLUME_COUNT = 1;
;  2612     2607  4              HOMEVBN = .VBN;
;  2613     2608  4              RETURN;
;  2614     2609  3              END;
;  2615     2610  2      END;
;  2616     2611  2
;  2617     2612  2
;  2618     2613  2  ! No good home block found.  Report failure.
;  2619     2614  2  !
;  2620     2615  2  SIGNAL(VERIFY$_FINDHOME, 1, .RVN, .OLD_STATUS);
;  2621     2616  1  END;
```

```
                                00FC 00000 READ_HOMEBLOCK:
                                           .WORD   Save R2,R3,R4,R5,R6,R7                  ; 2504
                    57 00000000G   00  9E 00002    MOVAB   LIB$SIGNAL, R7
                    56 00000000G   EF  9E 00009    MOVAB   CHECKSUM2, R6
                    55 00000000'   EF  9E 00010    MOVAB   BUFFER, R5
                    54             2C  D0 00017    MOVL    #44, OLD_STATUS                 ; 2545
                    52             02  D0 0001A    MOVL    #2, VBN                         ; 2546
                                   7E  7C 0001D 1$: CLRQ    -(SP)                           ; 2554
                                   7E  D4 0001F    CLRL    -(SP)
                                   52  DD 00021    PUSHL   VBN
                    7E      0200   8F  3C 00023    MOVZWL  #512, -(SP)
                                   55  DD 00028    PUSHL   R5
                                   7E  7C 0002A    CLRQ    -(SP)
                        00000000'  EF  9F 0002C    PUSHAB  IOSB
                                   31  DD 00032    PUSHL   #49
                              90   A5  DD 00034    PUSHL   CHANNEL
                                   7E  D4 00037    CLRL    -(SP)
                00000000G  00      0C  FB 00039    CALLS   #12, SYS$QIOW
                                   53  50 D0 00040 MOVL    R0, STATUS
                                   0A  53 E9 00043 BLBC    STATUS, 2$                      ; 2555
                    53 00000000'   EF  3C 00046    MOVZWL  IOSB, STATUS
                                   06  53 E8 0004D BLBS    STATUS, 3$                      ; 2557
                                   54  53 D0 00050 2$:    MOVL    STATUS, OLD_STATUS      ; 2559
                                 00DB  31 00053    BRW     8$
                    02          0D   A5  91 00056 3$:    CMPB    BUFFER+13, #2           ; 2562
                                   7F  12 00059    BNEQ    5$
                                08   A5  D5 0005C    TSTL    BUFFER+8                        ; 2563
                                   7A  13 0005F    BEQL    5$
                                0E   A5  B5 00061    TSTW    BUFFER+14                       ; 2564
                                   75  13 00064    BEQL    5$
                                10   A5  B5 00066    TSTW    BUFFER+16                       ; 2565
                                   70  13 00069    BEQL    5$
                                12   A5  B5 0006B    TSTW    BUFFER+18                       ; 2566
                                   6B  13 0006E    BEQL    5$
                                14   A5  B5 00070    TSTW    BUFFER+20                       ; 2567
                                   66  13 00073    BEQL    5$
                                16   A5  B5 00075    TSTW    BUFFER+22                       ; 2568
                                   61  13 00078    BEQL    5$
                                18   A5  D5 0007A    TSTL    BUFFER+24                       ; 2569
                                   5C  13 0007D    BEQL    5$
                                1C   A5  D5 0007F    TSTL    BUFFER+28                       ; 2570
                                   57  13 00082    BEQL    5$
                                20   A5  B5 00084    TSTW    BUFFER+32                       ; 2571
                                   52  13 00087    BEQL    5$
                                22   A5  B5 00089    TSTW    BUFFER+34                       ; 2572
                                   4D  13 0008C    BEQL    5$
                                   3A  DD 0008E    PUSHL   #58                             ; 2573
                                   55  DD 00090    PUSHL   R5
                    66             02  FB 00092    CALLS   #2, CHECKSUM2
                    43             50  E9 00095    BLBC    R0, 5$
                    7E      01FE   8F  3C 00098    MOVZWL  #510, -(SP)                     ; 2574
                                   55  DD 0009D    PUSHL   R5
                    66             02  FB 0009F    CALLS   #2, CHECKSUM2
```

```
                        36              50  E9  000A2          BLBC    R0, 5$
                        01        04    AC  D1  000A5          CMPL    RVN, #1                         ; 2580
                                        7E  12  000A9          BNEQ    7$
00000000'               EF              02  D0  000AB          MOVL    #2, STRUCTURE_LEVEL             ; 2583
00000000'               EF        28    A5  3C  000B2          MOVZWL  BUFFER+40, VOLUME_COUNT         ; 2584
                                        07  12  000BA          BNEQ    4$                              ; 2585
00000000'               EF              01  D0  000BC          MOVL    #1, VOLUME_COUNT
000000FF                8F  00000000'   EF  D1  000C3  4$:     CMPL    VOLUME_COUNT, #255              ; 2586
                                        59  1B  000CE          BLEQU   7$
                            00000000G   8F  DD  000D0          PUSHL   #VERIFY$_MAXVOLS
                        67              01  FB  000D6          CALLS   #1, LIB$SIGNAL
                                        4E  11  000D9          BRB     7$                              ; 2588
                        01        04    AC  D1  000DB  5$:     CMPL    RVN, #1                         ; 2592
                                        50  12  000DF          BNEQ    8$
                0101    8F        0C    A5  B1  000E1          CMPW    BUFFER+12, #257                 ; 2593
                                        08  13  000E7          BEQL    6$
                0102    8F        0C    A5  B1  000E9          CMPW    BUFFER+12, #258
                                        40  12  000EF          BNEQ    8$
                                  08    A5  B5  000F1  6$:     TSTW    BUFFER+8                        ; 2594
                                        3B  13  000F4          BEQL    8$
                                  02    A5  D5  000F6          TSTL    BUFFER+2                        ; 2595
                                        36  13  000F9          BEQL    8$
                                  06    A5  B5  000FB          TSTW    BUFFER+6                        ; 2596
                                        31  13  000FE          BEQL    8$
                                        65  B5  00100          TSTW    BUFFER                          ; 2597
                                        2D  13  00102          BEQL    8$
                                        3A  DD  00104          PUSHL   #58                             ; 2598
                                        55  DD  00106          PUSHL   R5
                        66              02  FB  00108          CALLS   #2, CHECKSUM2
                        23              50  E9  0010B          BLBC    R0, 8$
                        7E        01FE  8F  3C  0010E          MOVZWL  #510, -(SP)                     ; 2599
                                        55  DD  00113          PUSHL   R5
                        66              02  FB  00115          CALLS   #2, CHECKSUM2
                        16              50  E9  00118          BLBC    R0, 8$
            00000000'   EF              01  D0  0011B          MOVL    #1, STRUCTURE_LEVEL             ; 2605
            00000000'   EF              01  D0  00122          MOVL    #1, VOLUME_COUNT                ; 2606
            00000000'   EF              52  D0  00129  7$:     MOVL    VBN, HOMEVBN                    ; 2607
                                            04  00130          RET                                     ; 2601
        FEE2        52          01  00000064    8F  F1  00131  8$:     ACBL    #100, #1, VBN, 1$       ; 2546
                                                54  DD  00138          PUSHL   OLD_STATUS              ; 2615
                                          04    AC  DD  0013D          PUSHL   RVN
                                                01  DD  00140          PUSHL   #1
                                    00000000G   8F  DD  00142          PUSHL   #VERIFY$_FINDHOME
                                67              04  FB  00148          CALLS   #4, LIB$SIGNAL
                                                04  0014B          RET                             ; 2616

; Routine Size:  332 bytes,    Routine Base:  CODE + 25B1
```

```
;  2623     2617  1 ROUTINE SCAN_INDEX: NOVALUE=
;  2624     2618  1
;  2625     2619  1 !++
;  2626     2620  1 !
;  2627     2621  1 ! FUNCTIONAL DESCRIPTION:
;  2628     2622  1 !       This routine scans the index files on all volumes of a volume set.
;  2629     2623  1 !
;  2630     2624  1 ! INPUT PARAMETERS:
;  2631     2625  1 !       NONE
;  2632     2626  1 !
;  2633     2627  1 ! IMPLICIT INPUTS:
;  2634     2628  1 !       NONE
;  2635     2629  1 !
;  2636     2630  1 ! OUTPUT PARAMETERS:
;  2637     2631  1 !       NONE
;  2638     2632  1 !
;  2639     2633  1 ! IMPLICIT OUTPUTS:
;  2640     2634  1 !       NONE
;  2641     2635  1 !
;  2642     2636  1 ! ROUTINE VALUE:
;  2643     2637  1 !       NONE
;  2644     2638  1 !
;  2645     2639  1 ! SIDE EFFECTS:
;  2646     2640  1 !       NONE
;  2647     2641  1 !
;  2648     2642  1 !--
;  2649     2643  1
;  2650     2644  2 BEGIN
;  2651     2645  2 INCR RVN FROM 1 TO .VOLUME_COUNT DO
;  2652     2646  2     BEGIN
;  2653     2647  2     LOCAL
;  2654     2648  2         STATUS,                             ! Status variable
;  2655     2649  2         VBN;                                ! Current index file VBN
;  2656     2650  2
;  2657     2651  2
;  2658     2652  2     ! Access the index file.
;  2659     2653  2     !
;  2660     2654  2     CH$FILL(0, FIB$C_LENGTH, FIB);
;  2661     2655  2     FIB[FIB$L_ACCTL] = .ACCTL[.RVN-1];
;  2662     2656  2     FIB[FIB$W_FID_NUM] = FID$C_INDEXF;
;  2663     2657  2     FIB[FIB$W_FID_SEQ] = FID$C_INDEXF;
;  2664     2658  2     FIB[FIB$W_FID_RVN] = .RVN;
;  2665   P 2659  2     STATUS = $QIOW(
;  2666   P 2660  2         FUNC=IO$_ACCESS OR IO$M_ACCESS,
;  2667   P 2661  2         CHAN=.CHANNEL,
;  2668   P 2662  2         IOSB=IOSB,
;  2669     2663  2         P1=FIB_DESC);
;  2670     2664  2     IF .STATUS THEN STATUS = .IOSB[0];
;  2671     2665  2     IF NOT .STATUS THEN SIGNAL(VERIFY$_OPENINDEX, 1, .RVN, .STATUS);
;  2672     2666  2
;  2673     2667  2
;  2674     2668  2     ! Loop for all headers in the index file.  Read multiple blocks into a
;  2675     2669  2     ! buffer and process them one at a time.
;  2676     2670  2     !
;  2677     2671  2     VBN = .HEADER_OFFSET[.RVN-1] + 1;
;  2678     2672  3     UNTIL .VBN GTRU .EOF[.RVN-1] DO
;  2679     2673  4         BEGIN
```

```
;  2680          2674  4          LOCAL
;  2681          2675  4              HEADER:     REF BBLOCK,     ! Pointer to current header
;  2682          2676  4              READ_COUNT;                 ! Count of blocks to read
;  2683          2677  4
;  2684          2678  4
;  2685          2679  4          ! Establish the count of blocks to read and execute the read.
;  2686          2680  4          !
;  2687          2681  4          READ_COUNT = MINU(INDEX_BUF_COUNT, .EOF[.RVN-1] + 1 - .VBN);
;  2688        P 2682  4          STATUS = $QIOW(
;  2689        P 2683  4              FUNC=IO$_READVBLK,
;  2690        P 2684  4              CHAN=.CHANNEL,
;  2691        P 2685  4              IOSB=IOSB,
;  2692        P 2686  4              P1=BUFFER,
;  2693        P 2687  4              P2=512 * .READ_COUNT,
;  2694          2688  4              P3=.VBN);
;  2695          2689  4          IF .STATUS THEN STATUS = .IOSB[0];
;  2696          2690  4
;  2697          2691  4
;  2698          2692  4          ! If an error occurred, read each block separately, reporting any
;  2699          2693  4          ! errors.  Each header that reads with an error is deleted, since
;  2700           694  4          ! it cannot be trusted.
;  2701          2695  4          !
;  2702          2696  4          IF NOT .STATUS
;  2703          2697  4          THEN
;  2704          2698  5              BEGIN
;  2705          2699  5              INCR XVBN FROM 0 TO .READ_COUNT-1 DO
;  2706          2700  6                  BEGIN
;  2707          2701  6                  LOCAL
;  2708          2702  6                      HEADER:     REF BBLOCK,             ! Pointer to header
;  2709          2703  6                      FILE_NUMBER,                        ! Current file number
;  2710          2704  6                      FILE_ID:    BBLOCK[FID$C_LENGTH];   ! Current file ID
;  2711          2705  6
;  2712          2706  6
;  2713          2707  6                  ! Execute the read.
;  2714          2708  6                  !
;  2715          2709  6                  HEADER = BUFFER + .XVBN * 512;
;  2716        P 2710  6                  STATUS = $QIOW(
;  2717        P 2711  6                      FUNC=IO$_READVBLK,
;  2718        P 2712  6                      CHAN=.CHANNEL,
;  2719        P 2713  6                      IOSB=IOSB,
;  2720        P 2714  6                      P1=.HEADER,
;  2721        P 2715  6                      P2=512,
;  2722          2716  6                      P3=.VBN + .XVBN);
;  2723          2717  6                  IF .STATUS THEN STATUS = .IOSB[0];
;  2724          2718  6                  IF NOT .STATUS
;  2725          2719  6                  THEN
;  2726          2720  7                      BEGIN
;  2727          2721  7
;  2728          2722  7                      ! Get a clean file ID.
;  2729          2723  7                      !
;  2730          2724  7                      FILE_NUMBER = .VBN + .XVBN - .HEADER_OFFSET[.RVN-1];
;  2731          2725  7                      FILE_ID[FID$W_NUM] = .FILE_NUMBER;
;  2732          2726  7                      FILE_ID[FID$B_NMX] = .FILE_NUMBER<16,8>;
;  2733          2727  7                      IF .STRUCTURE_LEVEL EQL 2
;  2734          2728  7                          THEN FILE_ID[FID$W_SEQ] = .HEADER[FH2$W_FID_SEQ]
;  2735          2729  7                          ELSE FILE_ID[FID$W_SEQ] = .HEADER[FH1$W_FID_SEQ];
;  2736          2730  7                      FILE_ID[FID$B_RVN] = .RVN;
```

```
;  2737     2731  7
;  2738     2732  7
;  2739     2733  7                    ! Issue the error.
;  2740     2734  7                    !
;  2741     2735  7                    HEADER_ERROR(
;  2742     2736  7                        VERIFY$_READHEADER, FILE_ID, .HEADER,
;  2743     2737  7                        .STATUS);
;  2744     2738  7
;  2745     2739  7
;  2746     2740  7                    ! If we are repairing damage, rewrite the header with a
;  2747     2741  7                    ! deleted header.  In either case, ensure that the header
;  2748     2742  7                    ! is invalidated so that we will not process it.
;  2749     2743  7                    !
;  2750     2744  7                    HEADER[FH2$B_STRUCLEV] = 0;
;  2751     2745  7                    IF DO_REPAIR(NO_CONFIRM)
;  2752     2746  7                    THEN
;  2753     2747  7                        DELETE_HEADER(FILE_ID, .HEADER)
;  2754     2748  7                    ELSE
;  2755     2749  7                        BITVECTOR[.IMAP[.RVN-1], .FILE_NUMBER-1] = FALSE;
;  2756     2750  6                    END;
;  2757     2751  5                END;
;  2758     2752  4            END;
;  2759     2753  4
;  2760     2754  4
;  2761     2755  4            ! For each header, verify that it is a valid file header.
;  2762     2756  4            ! If it is, process it.
;  2763     2757  4            !
;  2764     2758  4            HEADER = BUFFER;
;  2765     2759  4            INCR XVBN FROM 0 TO .READ_COUNT-1 DO
;  2766     2760  5                BEGIN
;  2767     2761  5                LOCAL
;  2768     2762  5                    IDENT_AREA:     REF BBLOCK,     ! Pointer to ident area
;  2769     2763  5                    MAP_AREA:       REF BBLOCK,     ! Pointer to map area
;  2770     2764  5                    EXT_SEQ,                        ! Current extension sequence
;  2771     2765  5                    FILE_NUMBER,                    ! Current file number
;  2772     2766  5                    FILE_ID:        BBLOCK[FID$C_LENGTH];   ! Current file ID
;  2773     2767  5
;  2774     2768  5
;  2775     2769  5                ! Get a clean file ID.
;  2776     2770  5                !
;  2777     2771  5                FILE_NUMBER = .VBN + .XVBN - .HEADER_OFFSET[.RVN-1];
;  2778     2772  5                FILE_ID[FID$W_NUM] = .FILE_NUMBER;
;  2779     2773  5                FILE_ID[FID$B_NMX] = .FILE_NUMBER<16,8>;
;  2780     2774  5                IF .STRUCTURE_LEVEL EQL 2
;  2781     2775  5                    THEN FILE_ID[FID$W_SEQ] = .HEADER[FH2$W_FID_SEQ]
;  2782     2776  5                    ELSE FILE_ID[FID$W_SEQ] = .HEADER[FH1$W_FID_SEQ];
;  2783     2777  5                FILE_ID[FID$B_RVN] = .RVN;
;  2784     2778  5
;  2785     2779  5
;  2786     2780  5                ! Validate the header.  In ODS-2, a valid header is to be taken as
;  2787     2781  5                ! valid even if the index file bitmap shows it as available.  In
;  2788     2782  5                ! ODS-1, a header corresponding to a clear index file bitmap bit
;  2789     2783  5                ! is not to be examined.
;  2790     2784  5                !
;  2791     2785  5                STATUS = VERIFY_HEADER(.HEADER, FILE_ID);
;  2792     2786  5                IF
;  2793     2787  5                    .STRUCTURE_LEVEL EQL 1 AND
```

```
;  2794     2788  5                NOT .BITVECTOR[.IMAP[.RVN-1], .FILE_NUMBER-1]
;  2795     2789  5            THEN
;  2796     2790  5                STATUS = FALSE;
;  2797     2791  5
;  2798     2792  5
;  2799     2793  5            IF .STATUS
;  2800     2794  5            THEN
;  2801     2795  6                BEGIN
;  2802     2796  6
;  2803     2797  6                ! Header is valid.
;  2804     2798  6                !
;  2805     2799  6                IDENT_AREA = .HEADER + .HEADER[FH1$B_IDOFFSET]*2;
;  2806     2800  6                MAP_AREA = .HEADER + .HEADER[FH1$B_MPOFFSET]*2;
;  2807     2801  6
;  2808     2802  6
;  2809     2803  6                IF NOT .DUAL_ALLOC_PASS
;  2810     2804  6                THEN
;  2811     2805  7                    BEGIN
;  2812     2806  7                    IF .QUAL[QUAL_LIST]
;  2813     2807  7                    THEN
;  2814     2808  8                        BEGIN
;  2815     2809  8                        LOCAL
;  2816     2810  8                            FILENAME:   VECTOR[FI2$S_FILENAME + FI2$S_FILENAMEXT + 1, BYTE],
;  2817     2811  8                            LENGTH,
;  2818     2812  8                            OWNER,
;  2819     2813  8                            EXTENSION_SEGMENT;
;  2820     2814  8
;  2821     2815  8
;  2822     2816  8                        IF .STRUCTURE_LEVEL EQL 2
;  2823     2817  8                        THEN
;  2824     2818  9                            BEGIN
;  2825     2819  9                            CH$COPY(
;  2826     2820  9                                FI2$S_FILENAME, IDENT_AREA[FI2$T_FILENAME],
;  2827     2821  9                                %C' ',
;  2828     2822  9                                FI2$S_FILENAME + FI2$S_FILENAMEXT + 1, FILENAME);
;  2829     2823  9
;  2830     2824  9                            IF (.HEADER[FH2$B_MPOFFSET] - .HEADER[FH2$B_IDOFFSET]) * 2
;  2831     2825  9                            GEQU $BYTEOFFSET(FI2$T_FILENAMEXT) + FI2$S_FILENAMEXT
;  2832     2826  9                            THEN
;  2833     2827  9                                CH$MOVE(
;  2834     2828  9                                    FI2$S_FILENAMEXT,
;  2835     2829  9                                    IDENT_AREA[FI2$T_FILENAMEXT],
;  2836     2830  9                                    FILENAME[FI2$S_FILENAME]);
;  2837     2831  9
;  2838     2832  9                            LENGTH =
;  2839     2833  9                                CH$FIND_CH(FI2$S_FILENAME + FI2$S_FILENAMEXT + 1, FILENAME, %C' ')
;  2840     2834  9                                - FILENAME;
;  2841     2835  9
;  2842     2836  9                            OWNER = .HEADER[FH2$L_FILEOWNER];
;  2843     2837  9                            EXTENSION_SEGMENT = .HEADER[FH2$W_SEG_NUM];
;  2844     2838  9                            END
;  2845     2839  8                    ELSE
;  2846     2840  9                        BEGIN
;  2847     2841  9                        LENGTH = MAKE_STRING(
;  2848     2842  9                            IDENT_AREA[FI1$W_FILENAME] - $BYTEOFFSET(NMB$W_NAME),
;  2849     2843  9                            FILENAME);
;  2850     2844  9
```

```
;  2851          2845  9                    OWNER = .HEADER[FH1$B_UICMEMBER];
;  2852          2846  9                    OWNER<16,16> = .HEADER[FH1$B_UICGROUP];
;  2853          2847  9                    EXTENSION_SEGMENT = .MAP_AREA[FM1$B_EX_SEGNUM];
;  2854          2848  8                    END;
;  2855          2849  8
;  2856        P 2850  8                FAO_(
;  2857        P 2851  8                    '(!8ZL,!5ZL,!3ZL)  !86AF  !%U',
;  2858        P 2852  8                    .FILE_NUMBER,
;  2859        P 2853  8                    .FILE_ID[FID$W_SEQ],
;  2860        P 2854  8                    .FILE_ID[FID$B_RVN],
;  2861        P 2855  8                    .LENGTH, FILENAME,
;  2862          2856  8                    .OWNER);
;  2863          2857  8
;  2864          2858  8                IF .EXTENSION_SEGMENT NEQ 0
;  2865          2859  8                THEN
;  2866          2860  8                    FAO_('  extension !UL', .EXTENSION_SEGMENT);
;  2867          2861  8
;  2868          2862  8                EOL();
;  2869          2863  7                END;
;  2870          2864  7
;  2871          2865  7
;  2872          2866  7            ! Make sure the index file bitmap indicates that the header
;  2873          2867  7            ! is valid.  Remember the file sequence number and the
;  2874          2868  7            ! back link FID.
;  2875          2869  7            !
;  2876          2870  7            BITVECTOR[.IMAP[.RVN-1], .FILE_NUMBER-1] = TRUE;
;  2877          2871  7            VECTOR[.SEQMAP[.RVN-1], .FILE_NUMBER-1 ;,WORD] = .FILE_ID[FID$W_SEQ];
;  2878          2872  7            IF .STRUCTURE_LEVEL EQL 2
;  2879          2873  7            THEN
;  2880          2874  8                BEGIN
;  2881          2875  8                LOCAL
;  2882          2876  8                    BACK_ID:    REF BBLOCK;
;  2883          2877  8
;  2884          2878  8                BACK_ID = VECTOR[.BACKMAP[.RVN-1], (.FILE_NUMBER-1)*3 ;,WORD];
;  2885          2879  8                BACK_ID[FID$W_NUM] = .HEADER[FH2$W_BK_FIDNUM];
;  2886          2880  8                BACK_ID[FID$W_SEQ] = .HEADER[FH2$W_BK_FIDSEQ];
;  2887          2881  8                BACK_ID[FID$W_RVN] = .HEADER[FH2$W_BK_FIDRVN];
;  2888          2882  8                IF .BACK_ID[FID$B_RVN] EQL 0 THEN BACK_ID[FID$B_RVN] = .RVN;
;  2889          2883  7                END;
;  2890          2884  6            END;
;  2891          2885  6
;  2892          2886  6
;  2893          2887  6            ! Other processing executed only if not an extension header.
;  2894          2888  6            !
;  2895          2889  6            IF
;  2896          2890  7                BEGIN
;  2897          2891  7                IF .STRUCTURE_LEVEL EQL 2
;  2898          2892  7                    THEN .HEADER[FH2$W_SEG_NUM] EQL 0
;  2899          2893  7                    ELSE .MAP_AREA[FM1$B_EX_SEGNUM] EQL 0
;  2900          2894  7                END
;  2901          2895  6            THEN
;  2902          2896  7                BEGIN
;  2903          2897  7                LOCAL
;  2904          2898  7                    FAT:    REF BBLOCK,
;  2905          2899  7                    FCH:    REF BBLOCK;
;  2906          2900  7
;  2907          2901  7
```

```
;  2908     2902  7                    IF NOT .DUAL_ALLOC_PASS
;  2909     2903  7                    THEN
;  2910     2904  8                        BEGIN
;  2911     2905  8
;  2912     2906  8                        ! Directory entry should point to this file.
;  2913     2907  8                        !
;  2914     2908  8                        BITVECTOR[.LOSTMAP[.RVN-1], .FILE_NUMBER-1] = TRUE;
;  2915     2909  8
;  2916     2910  8
;  2917     2911  8                        ! Get file characteristics and attributes pointer.
;  2918     2912  8                        !
;  2919     2913  8                        IF .STRUCTURE_LEVEL EQL 2
;  2920     2914  8                        THEN
;  2921     2915  9                            BEGIN
;  2922     2916  9                            FCH = HEADER[FH2$L_FILECHAR];
;  2923     2917  9                            FAT = HEADER[FH2$W_RECATTR];
;  2924     2918  9                            END
;  2925     2919  8                        ELSE
;  2926     2920  9                            BEGIN
;  2927     2921  9                            FCH = HEADER[FH1$W_FILECHAR];
;  2928     2922  9                            FAT = HEADER[FH1$W_RECATTR];
;  2929     2923  8                            END;
;  2930     2924  8
;  2931     2925  8
;  2932     2926  8                        ! Report file marked for delete.
;  2933     2927  8                        !
;  2934     2928  8                        IF .FCH[FCH$V_MARKDEL]
;  2935     2929  8                        THEN
;  2936     2930  9                            BEGIN
;  2937     2931  9                            HEADER_ERROR(VERIFY$_DELHEADER, FILE_ID, .HEADER);
;  2938     2932  9                            IF DO_REPAIR()
;  2939     2933  9                            THEN
;  2940     2934 10                                BEGIN
;  2941     2935 10                                ENTER_WORK(WRK_K_DELETE, FILE_ID);
;  2942     2936 10                                VECTOR[.SEQMAP[.RVN-1], .FILE_NUMBER-1 ;,WORD] = -1;
;  2943     2937  9                                END;
;  2944     2938  8                            END;
;  2945     2939  8
;  2946     2940  8
;  2947     2941  8                        ! Report file deaccess locked.
;  2948     2942  8                        !
;  2949     2943  8                        IF .FCH[FCH$V_LOCKED]
;  2950     2944  8                        THEN
;  2951     2945  9                            BEGIN
;  2952     2946  9                            HEADER_ERROR(VERIFY$_LOCKHEADER, FILE_ID, .HEADER);
;  2953     2947 10                            IF (STATUS = DO_REPAIR(ALLOW_DELETE))
;  2954     2948  9                            THEN
;  2955     2949 10                                BEGIN
;  2956     2950 10                                IF .STATUS<1,1>
;  2957     2951 10                                THEN
;  2958     2952 11                                    BEGIN
;  2959     2953 11                                    ENTER_WORK(WRK_K_DELETE, FILE_ID);
;  2960     2954 11                                    VECTOR[.SEQMAP[.RVN-1], .FILE_NUMBER-1 ;,WORD] = -1;
;  2961     2955 11                                    END
;  2962     2956 10                                ELSE
;  2963     2957 11                                    BEGIN
;  2964     2958 11                                    FCH[FCH$V_LOCKED] = FALSE;
```

```
;  2965     2959 11                        WRITE_HEADER(FILE_ID, .HEADER);
;  2966     2960 10                        END;
;  2967     2961  9                    END;
;  2968     2962  8                END;
;  2969     2963  8
;  2970     2964  8
;  2971     2965  8                ! Report file containing suspected bad blocks.
;  2972     2966  8                !
;  2973     2967  8                IF .FCH[FCH$V_BADBLOCK]
;  2974     2968  8                THEN
;  2975     2969  8                    HEADER_ERROR(VERIFY$_BBLHEADER, FILE_ID, .HEADER);
;  2976     2970  8
;  2977     2971  8
;  2978     2972  8                ! Remember directory bit.
;  2979     2973  8                !
;  2980     2974  8                IF
;  2981     2975  8                    .FILE_NUMBER NEQ FID$C_MFD
;  2982     2976  8                AND
;  2983     2977  9                    BEGIN
;  2984     2978  9                    IF .STRUCTURE_LEVEL EQL 2
;  2985     2979  9                    THEN
;  2986     2980  9                        .HEADER[FM2$V_DIRECTORY]
;  2987     2981  9                    ELSE
;  2988     2982  9                        .FAT[FAT$B_RTYPE] EQL FAT$C_FIXED AND
;  2989     2983  9                        .FAT[FAT$W_RSIZE] EQL NMBSC_DIRENTRY AND
;  2990     2984  9                        .IDENT_AREA[FI1$W_FILETYPE] EQL %RAD50_11 'DIR'
;  2991     2985  9                    END
;  2992     2986  8                THEN
;  2993     2987  8                    BITVECTOR[.DIRMAP[.RVN-1], .FILE_NUMBER-1] = TRUE;
;  2994     2988  8
;  2995     2989  8                ! Check creation date.
;  2996     2990  8                !
;  2997     2991  8                CHECK_DATE(
;  2998     2992  8                    IDENT_AREA[FI2$Q_CREDATE],
;  2999     2993  8                    IDENT_AREA[FI1$T_CREDATE],
;  3000     2994  8                    VERIFY$_FUTCREDAT,
;  3001     2995  8                    FILE_ID,
;  3002     2996  8                    .HEADER);
;  3003     2997  8
;  3004     2998  8
;  3005     2999  8                ! Check revision date.
;  3006     3000  8                !
;  3007     3001  8                CHECK_DATE(
;  3008     3002  8                    IDENT_AREA[FI2$Q_REVDATE],
;  3009     3003  8                    IDENT_AREA[FI1$T_REVDATE],
;  3010     3004  8                    VERIFY$_FUTREVDAT,
;  3011     3005  8                    FILE_ID,
;  3012     3006  8                    .HEADER);
;  3013     3007  8
;  3014     3008  8
;  3015     3009  8                ! Check backup date.
;  3016     3010  8                !
;  3017     3011  8                IF .STRUCTURE_LEVEL EQL 2
;  3018     3012  8                THEN
;  3019     3013  8                    CHECK_DATE(
;  3020     3014  8                        IDENT_AREA[FI2$Q_BAKDATE],
;  3021     3015  8
```

```
;  3022     3016  8                    0,
;  3023     3017  8                    VERIFY$_FUTBAKL ,,
;  3024     3018  8                    FILE_ID,
;  3025     3019  8                    .HEADER);
;  3026     3020  7            END;
;  3027     3021  7
;  3028     3022  7
;  3029     3023  7        ! Process map area.
;  3030     3024  7        !
;  3031     3025  7        TOTAL_SIZE = 0;
;  3032     3026  7        MAP_PROCESS(.HEADER, .RVN, .HEADER, FILE_ID, +1);
;  3033     3027  7
;  3034     3028  7
;  3035     3029  7        ! If file has extension headers, run down the chain
;  3036     3030  7        ! checking them.
;  3037     3031  7        !
;  3038     3032  7        EXT_SEQ = 0;
;  3039     3033  7        IF
;  3040     3034  8            BEGIN
;  3041     3035  8            IF .STRUCTURE_LEVEL EQL 2
;  3042     3036  8                THEN .HEADER[FH2$W_EX_FIDNUM] NEQ 0
;  3043     3037  8                ELSE .MAP_AREA[FM1$W_EX_FILNUM] NEQ 0
;  3044     3038  8            END
;  3045     3039  7        THEN
;  3046     3040  8            BEGIN
;  3047     3041  8            LOCAL
;  3048     3042  8                EXT_HDR:            REF BBLOCK,
;  3049     3043  8                EXT_MAP_AREA:       REF BBLOCK,
;  3050     3044  8                EXT_RVN,
;  3051     3045  8                PREV_FILE_ID:       BBLOCK[FID$C_LENGTH];
;  3052     3046  8
;  3053     3047  8
;  3054     3048  8            EXT_HDR = .HEADER;
;  3055     3049  8            EXT_MAP_AREA = .MAP_AREA;
;  3056     3050  8            EXT_RVN = .RVN;
;  3057     3051  8            PREV_FILE_ID[FID$W_NUM] = .FILE_ID[FID$W_NUM];
;  3058     3052  8            PREV_FILE_ID[FID$W_SEQ] = .FILE_ID[FID$W_SEQ];
;  3059     3053  8            PREV_FILE_ID[FID$W_RVN] = .FILE_ID[FID$W_RVN];
;  3060     3054  8            WHILE
;  3061     3055  9                BEGIN
;  3062     3056  9                IF .STRUCTURE_LEVEL EQL 2
;  3063     3057  9                    THEN .EXT_HDR[FH2$W_EX_FIDNUM] NEQ 0
;  3064     3058  9                    ELSE .EXT_MAP_AREA[FM1$W_EX_FILNUM] NEQ 0
;  3065     3059  9                END
;  3066     3060  8            DO
;  3067     3061  9                BEGIN
;  3068     3062  9                LOCAL
;  3069     3063  9                    EXT_FILE_NUMBER,
;  3070     3064  9                    EXT_FILE_ID:    BBLOCK[FID$C_LENGTH];
;  3071     3065  9
;  3072     3066  9                ! Get clean file number and RVN.
;  3073     3067  9                !
;  3074     3068  9                IF .STRUCTURE_LEVEL EQL 2
;  3075     3069  9                THEN
;  3076     3070  9                    BEGIN
;  3077     3071 10                    EXT_FILE_NUMBER = .EXT_HDR[FH2$W_EX_FIDNUM];
;  3078     3072 10
```

```
;  3079     3073 10                        EXT_FILE_NUMBER<16,8> = .EXT_HDR[FH2$B_EX_FIDNMX];
;  3080     3074 10                        EXT_FILE_ID[FID$W_NUM] = .EXT_HDR[FH2$W_EX_FIDNUM];
;  3081     3075 10                        EXT_FILE_ID[FID$W_SEQ] = .EXT_HDR[FH2$W_EX_FIDSEQ];
;  3082     3076 10                        EXT_FILE_ID[FID$W_RVN] = .EXT_HDR[FH2$W_EX_FIDRVN];
;  3083     3077 10                        END
;  3084     3078  9                    ELSE
;  3085     3079 10                        BEGIN
;  3086     3080 10                        EXT_FILE_NUMBER = .EXT_MAP_AREA[FM1$W_EX_FILNUM];
;  3087     3081 10                        EXT_FILE_ID[FID$W_NUM] = .EXT_MAP_AREA[FM1$W_EX_FILNUM];
;  3088     3082 10                        EXT_FILE_ID[FID$W_SEQ] = .EXT_MAP_AREA[FM1$W_EX_FILSEQ];
;  3089     3083 10                        EXT_FILE_ID[FID$W_RVN] = 1;
;  3090     3084  9                        END;
;  3091     3085  9                    IF .EXT_FILE_ID[FID$B_RVN] EQL 0 THEN EXT_FILE_ID[FID$B_RVN] = .EXT_RVN;
;  3092     3086  9                    EXT_RVN = .EXT_FILE_ID[FID$B_RVN];
;  3093     3087  9
;  3094     3088  9                    ! If extension linkage points to CONTIN.SYS, exit
;  3095     3089  9                    ! the loop, since following segments of the file
;  3096     3090  9                    ! exist as part of a loosely coupled volume set.
;  3097     3091  9                    !
;  3098     3092  9                    IF
;  3099     3093  9                        .STRUCTURE_LEVEL EQL 2 AND
;  3100     3094  9                        .EXT_FILE_NUMBER EQL FID$C_CONTIN AND
;  3101     3095  9                        .EXT_FILE_ID[FID$W_SEQ] EQL FID$C_CONTIN
;  3102     3096  9                    THEN
;  3103     3097  9                        EXITLOOP;
;  3104     3098  9
;  3105     3099  9
;  3106     3100  9                    ! Validity check extension file ID.  If this fails,
;  3107     3101  9                    ! clear the extension linkage and exit the loop.
;  3108     3102  9                    !
;  3109     3103  9                    IF
;  3110     3104  9                        BEGIN
;  3111     3105 10                        IF .EXT_RVN GTRU .VOLUME_COUNT
;  3112     3106 10                        THEN
;  3113     3107 10                            TRUE
;  3114     3108 10                        ELSE
;  3115     3109 10                            .EXT_FILE_NUMBER-1 GTRU .MAXFILIDX[.EXT_RVN-1]
;  3116     3110 10                        END
;  3117     3111 10                    THEN
;  3118     3112  9                        BEGIN
;  3119     3113 10                        IF NOT .DUAL_ALLOC_PASS
;  3120     3114 10                        THEN
;  3121     3115 10                            BEGIN
;  3122     3116 11                            HEADER_ERROR(VERIFY$_INVEXTFID, FILE_ID, .HEADER);
;  3123     3117 11                            IF DO_REPAIR()
;  3124     3118 11                            THEN
;  3125     3119 11                                CLEAR_EXT_FID(FILE_ID, .HEADER);
;  3126     3120 11                            END;
;  3127     3121 10                        EXITLOOP;
;  3128     3122 10                        END;
;  3129     3123  9
;  3130     3124  9
;  3131     3125  9                    ! Read extension file header.  If this fails,
;  3132     3126  9                    ! exit the loop.
;  3133     3127  9                    !
;  3134     3128  9                    IF NOT READ_HEADER(EXT_FILE_ID, BUFFER_2)
;  3135     3129  9
```

```
;  3136     3130  9                    THEN
;  3137     3131 10                        BEGIN
;  3138     3132 10                        IF NOT .DUAL_ALLOC_PASS
;  3139     3133 10                        THEN
;  3140     3134 11                            BEGIN
;  3141     3135 11                            IF DO_REPAIR()
;  3142     3136 11                            THEN
;  3143     3137 11                                IF READ_HEADER(PREV_FILE_ID, BUFFER_2)
;  3144     3138 11                                THEN
;  3145     3139 11                                    CLEAR_EXT_FID(PREV_FILE_ID, BUFFER_2);
;  3146     3140 10                            END;
;  3147     3141 10                        EXITLOOP;
;  3148     3142  9                        END;
;  3149     3143  9
;  3150     3144  9
;  3151     3145  9                    ! Adjust pointers to new header.
;  3152     3146  9                    !
;  3153     3147  9                    EXT_HDR = BUFFER_2;
;  3154     3148  9                    EXT_MAP_AREA = .EXT_HDR + .EXT_HDR[FH1$B_MPOFFSET]*2;
;  3155     3149  9                    EXT_SEQ = .EXT_SEQ + 1;
;  3156     3150  9
;  3157     3151  9
;  3158     3152  9                    ! Check segment number.  If this check fails,
;  3159     3153  9                    ! clear the forward link that points to the bad
;  3160     3154  9                    ! header and exit the loop.
;  3161     3155  9                    !
;  3162     3156  9                    IF
;  3163     3157 10                        BEGIN
;  3164     3158 10                        IF .STRUCTURE_LEVEL EQL 2
;  3165     3159 10                            THEN .EXT_HDR[FH2$W_SEG_NUM] NEQ .EXT_SEQ
;  3166     3160 10                            ELSE .EXT_MAP_AREA[FM1$B_EX_SEGNUM] NEQ .EXT_SEQ
;  3167     3161 10                        END
;  3168     3162  9                    THEN
;  3169     3163 10                        BEGIN
;  3170     3164 10                        IF NOT .DUAL_ALLOC_PASS
;  3171     3165 10                        THEN
;  3172     3166 11                            BEGIN
;  3173     3167 11                            HEADER_ERROR(VERIFY$_INVEXTHDR, EXT_FILE_ID, .EXT_HDR);
;  3174     3168 11                            IF DO_REPAIR()
;  3175     3169 11                            THEN
;  3176     3170 12                                BEGIN
;  3177     3171 12                                IF READ_HEADER(PREV_FILE_ID, BUFFER_2)
;  3178     3172 12                                THEN
;  3179     3173 12                                    CLEAR_EXT_FID(PREV_FILE_ID, BUFFER_2);
;  3180     3174 11                                END;
;  3181     3175 10                            END;
;  3182     3176 10                        EXITLOOP;
;  3183     3177  9                        END;
;  3184     3178  9
;  3185     3179  9
;  3186     3180  9                    ! Remember previous file ID in case we need to
;  3187     3181  9                    ! clear the forward link.
;  3188     3182  9                    !
;  3189     3183  9                    PREV_FILE_ID[FID$W_NUM] = .EXT_FILE_ID[FID$W_NUM];
;  3190     3184  9                    PREV_FILE_ID[FID$W_SEQ] = .EXT_FILE_ID[FID$W_SEQ];
;  3191     3185  9                    PREV_FILE_ID[FID$W_RVN] = .EXT_FILE_ID[FID$W_RVN];
;  3192     3186  9
```

```
;  3193     3187  9
;  3194     3188  9                    ! Do map area processing.
;  3195     3189  9                    !
;  3196     3190  9                    MAP_PROCESS(.EXT_HDR, .EXT_RVN, .HEADER, FILE_ID, 0);
;  3197     3191  9
;  3198     3192  9
;  3199     3193  9                    IF NOT .DUAL_ALLOC_PASS
;  3200     3194  9                    THEN
;  3201     3195 10                        BEGIN
;  3202     3196 10
;  3203     3197 10                        ! Check extension header back link.  It should
;  3204     3198 10                        ! point to primary header.
;  3205     3199 10                        !
;  3206     3200 10                        IF .STRUCTURE_LEVEL EQL 2
;  3207     3201 10                        THEN
;  3208     3202 11                            BEGIN
;  3209     3203 11                            LOCAL
;  3210     3204 11                                BCK_RVN;
;  3211     3205 11
;  3212     3206 11                            BCK_RVN = .EXT_HDR[FH2$B_BK_FIDRVN];
;  3213     3207 11                            IF .BCK_RVN EQL 0 THEN BCK_RVN = .EXT_FILE_ID[FID$B_RVN];
;  3214     3208 11                            IF
;  3215     3209 11                                .EXT_HDR[FH2$W_BK_FIDNUM] NEQ .FILE_ID[FID$W_NUM] OR
;  3216     3210 11                                .EXT_HDR[FH2$W_BK_FIDSEQ] NEQ .FILE_ID[FID$W_SEQ] OR
;  3217     3211 11                                .EXT_HDR[FH2$B_BK_FIDNMX] NEQ .FILE_ID[FID$B_NMX] OR
;  3218     3212 11                                .BCK_RVN NEQ .FILE_ID[FID$B_RVN]
;  3219     3213 11                            THEN
;  3220     3214 12                                BEGIN
;  3221     3215 12                                HEADER_ERROR(
;  3222     3216 12                                    VERIFY$_INVEXTBACK, EXT_FILE_ID, .EXT_HDR);
;  3223     3217 12                                IF DO_REPAIR()
;  3224     3218 12                                THEN
;  3225     3219 13                                    BEGIN
;  3226     3220 13                                    EXT_HDR[FH2$W_BK_FIDNUM] = .FILE_ID[FID$W_NUM];
;  3227     3221 13                                    EXT_HDR[FH2$W_BK_FIDSEQ] = .FILE_ID[FID$W_SEQ];
;  3228     3222 13                                    EXT_HDR[FH2$W_BK_FIDRVN] = .FILE_ID[FID$W_RVN];
;  3229     3223 13                                    IF .EXT_HDR[FH2$B_BK_FIDRVN] EQL .EXT_FILE_ID[FID$B_RVN]
;  3230     3224 13                                        THEN EXT_HDR[FH2$B_BK_FIDRVN] = 0;
;  3231     3225 13                                    WRITE_HEADER(EXT_FILE_ID, .EXT_HDR);
;  3232     3226 12                                    END;
;  3233     3227 11                                END;
;  3234     3228 10                        END;
;  3235     3229 10
;  3236     3230 10
;  3237     3231 10                    ! Mark as a referenced extension header.  If
;  3238     3232 10                    ! already marked, report multiple references.
;  3239     3233 10                    !
;  3240     3234 10                    IF TESTBITSS(BITVECTOR[.EXTMAP[.EXT_RVN-1], .EXT_FILE_NUMBER-1])
;  3241     3235 10                    THEN
;  3242     3236 10                        HEADER_ERROR(
;  3243     3237 10                            VERIFY$_MULTEXTHDR, EXT_FILE_ID, .EXT_HDR);
;  3244     3238 10
;  3245     3239 10
;  3246     3240 10                    ! Check file owner.  If not equal, should be
;  3247     3241 10                    ! corrected to charge space to rightful user.
;  3248     3242 10                    !
;  3249     3243 10                    IF
```

```
;  3250     3244 11                        BEGIN
;  3251     3245 11                        IF .STRUCTURE_LEVEL EQL 2
;  3252     3246 11                            THEN .EXT_HDR[FH2$L_FILEOWNER] NEQ .HEADER[FH2$L_FILEOWNER]
;  3253     3247 11                            ELSE .EXT_HDR[FH1$W_FILEOWNER] NEQ .HEADER[FH1$W_FILEOWNER]
;  3254     3248 11                        END
;  3255     3249 10                    THEN
;  3256     3250 11                        BEGIN
;  3257     3251 11                        HEADER_ERROR(VERIFY$_WRONGOWNER, EXT_FILE_ID, .EXT_HDR);
;  3258     3252 11                        IF DO_REPAIR()
;  3259     3253 11                        THEN
;  3260     3254 12                            BEGIN
;  3261     3255 12                            IF .STRUCTURE_LEVEL EQL 2
;  3262     3256 12                                THEN EXT_HDR[FH2$L_FILEOWNER] = .HEADER[FH2$L_FILEOWNER]
;  3263     3257 12                                ELSE EXT_HDR[FH1$W_FILEOWNER] = .HEADER[FH1$W_FILEOWNER];
;  3264     3258 12                            WRITE_HEADER(EXT_FILE_ID, .EXT_HDR);
;  3265     3259 11                            END;
;  3266     3260 10                        END;
;  3267     3261  9                    END;
;  3268     3262  8                END;
;  3269     3263  7            END;
;  3270     3264  7
;  3271     3265  7
;  3272     3266  7            IF NOT .DUAL_ALLOC_PASS
;  3273     3267  7            THEN
;  3274     3268  8                BEGIN
;  3275     3269  8
;  3276     3270  8                ! Check HIBLK and EFBLK in record attributes against
;  3277     3271  8                ! total size computed from map area.  Allow the error
;  3278     3272  8                ! for INDEXF.SYS, BITMAP.SYS, and BADBLK.SYS in ODS-1,
;  3279     3273  8                ! since HIBLK is not maintained for these files.
;  3280     3274  8                !
;  3281     3275  8                IF
;  3282     3276  8                    .TOTAL_SIZE NEQ ROT(.FAT[FAT$L_HIBLK], 16) AND
;  3283     3277  9                    (.STRUCTURE_LEVEL NEQ 1 OR .FILE_NUMBER GTRU FID$C_BADBLK)
;  3284     3278  8                THEN
;  3285     3279  9                    BEGIN
;  3286     3280  9                    HEADER_ERROR(VERIFY$_BADHIBLK, FILE_ID, .HEADER);
;  3287     3281  9                    IF DO_REPAIR()
;  3288     3282  9                    THEN
;  3289     3283 10                        BEGIN
;  3290     3284 10                        FAT[FAT$L_HIBLK] = ROT(.TOTAL_SIZE, 16);
;  3291     3285 10                        WRITE_HEADER(FILE_ID, .HEADER);
;  3292     3286  9                        END;
;  3293     3287  8                    END;
;  3294     3288  8
;  3295     3289  8
;  3296     3290  8                IF
;  3297     3291  8                    .FAT[FAT$L_EFBLK] NEQ 0 AND
;  3298     3292  9                    ROT(.FAT[FAT$L_EFBLK], 16) - (.FAT[FAT$W_FFBYTE] EQL 0)
;  3299     3293  8                    GTRU .TOTAL_SIZE
;  3300     3294  8                THEN
;  3301     3295  9                    BEGIN
;  3302     3296  9                    HEADER_ERROR(VERIFY$_BADEFBLK, FILE_ID, .HEADER);
;  3303     3297  9                    IF DO_REPAIR()
;  3304     3298  9                    THEN
;  3305     3299 10                        BEGIN
;  3306     3300 10                        FAT[FAT$L_EFBLK] = ROT(.TOTAL_SIZE + 1, 16);
```

```
;  3307     3301 10                    FAT[FAT$W_FFBYTE] = 0;
;  3308     3302 10                    WRITE_HEADER(FILE_ID, .HEADER);
;  3309     3303  9                    END;
;  3310     3304  8                END;
;  3311     3305  8
;  3312     3306  8
;  3313     3307  8                ! Read-check the file if requested.
;  3314     3308  8                !
;  3315     3309  8                IF .QUAL[QUAL_READ]
;  3316     3310  8                THEN
;  3317     3311  8                    IF .FILE_NUMBER GEQU FID$C_MFD
;  3318     3312  8                    THEN
;  3319     3313  8                        READ_CHECK(FILE_ID, .HEADER);
;  3320     3314  8
;  3321     3315  8
;  3322     3316  8                ! Update usage info.
;  3323     3317  8                !
;  3324     3318  8                IF .QUAL[QUAL_USAG]
;  3325     3319  8                THEN
;  3326     3320  9                    BEGIN
;  3327     3321  9                    IF .STRUCTURE_LEVEL EQL 2
;  3328     3322  9                        THEN VECTOR[.OWNER[.RVN-1], .FILE_NUMBER-1] = .HEADER[FH2$L_FILEOWNER]
;  3329     3323  9                        ELSE VECTOR[.OWNER[.RVN-1], .FILE_NUMBER-1] = .HEADER[FH1$B_UICGROUP]^16 + .
;  3330     3324  9                    VECTOR[.ALLOCATION[.RVN-1], .FILE_NUMBER-1] = .TOTAL_SIZE + .EXT_SEQ + 1;
;  3331     3325  9                    VECTOR[.USAGE[.RVN-1], .FILE_NUMBER-1] = ROT(.FAT[FAT$L_EFBLK], 16);
;  3332     3326  9                    IF
;  3333     3327  9                        .FAT[FAT$L_EFBLK] NEQ 0 AND
;  3334     3328  9                        .FAT[FAT$W_FFBYTE] EQL 0
;  3335     3329  9                    THEN
;  3336     3330  9                        VECTOR[.USAGE[.RVN-1], .FILE_NUMBER-1] =
;  3337     3331  9                            .VECTOR[.USAGE[.RVN-1], .FILE_NUMBER-1] - 1;
;  3338     3332  8                    END;
;  3339     3333  8
;  3340     3334  8
;  3341     3335  8                        ! Update quota info.
;  3342     3336  8                        !
;  3343     3337  8                        IF .QUOTA_ACTIVE AND .FILE_NUMBER GEQU FID$C_MFD
;  3344     3338  8                        THEN
;  3345     3339  8                            COUNT_QUOTA(
;  3346     3340  8                                .HEADER[FH2$L_FILEOWNER],
;  3347     3341  8                                0,
;  3348     3342  8                                .TOTAL_SIZE + .EXT_SEQ + 1);
;  3349     3343  7                        END;
;  3350     3344  7                END
;  3351     3345  6            ELSE
;  3352     3346  7                BEGIN
;  3353     3347  7
;  3354     3348  7                ! Extension header.  Just process the map area.
;  3355     3349  7                !
;  3356     3350  7                MAP_PROCESS(.HEADER, .RVN, .HEADER, FILE_ID, -1);
;  3357     3351  6                END;
;  3358     3352  6            END
;  3359     3353  5        ELSE
;  3360     3354  6            BEGIN
;  3361     3355  6            !
;  3362     3356  6            ! Invalid or deleted file header.  If it is marked busy in the
;  3363     3357  6            ! bitmap, complain and rewrite a valid deleted header.
```

```
;  3364     3358  6                    !
;  3365     3359  6                    IF NOT .DUAL_ALLOC_PASS
;  3366     3360  6                    THEN
;  3367     3361  7                        BEGIN
;  3368     3362  7                        IF .BITVECTOR[.IMAP[.RVN-1], .FILE_NUMBER-1]
;  3369     3363  7                        THEN
;  3370     3364  8                            BEGIN
;  3371     3365  8                            HEADER_ERROR(VERIFY$_BADHEADER, FILE_ID, .HEADER);
;  3372     3366  8                            IF DO_REPAIR()
;  3373     3367  8                            THEN
;  3374     3368  8                                DELETE_HEADER(FILE_ID, .HEADER);
;  3375     3369  7                            END;
;  3376     3370  6                        END;
;  3377     3371  5                    END;
;  3378     3372  5
;  3379     3373  5
;  3380     3374  5                HEADER = .HEADER + 512;
;  3381     3375  4                END;
;  3382     3376  4
;  3383     3377  4
;  3384     3378  4            VBN = .VBN + INDEX_BUF_COUNT;
;  3385     3379  4            END;
;  3386     3380  3
;  3387     3381  3
;  3388     3382  3            ! Deaccess the index file.
;  3389     3383  3            !
;  3390   P 3384  3            $QIOW(
;  3391   P 3385  3                FUNC=IO$_DEACCESS,
;  3392     3386  3                CHAN=.CHANNEL);
;  3393     3387  2            END;
;  3394     3388  1 END;


5A 33 21 2C 4C 5A 35 21 2C 4C 5A 38 21 28 1C 026FD P.ABK:  .ASCII  <28>\(!8ZL,!5ZL,!3ZL)  !86AF  !%U\
   55 25 21 20 20 46 41 36 38 21 20 20 29 4C 0270C
55 21 20 6E 6F 69 73 6E 65 74 78 65 20 20 0F 0271A P.ABL:  .ASCII  <15>\  extension !UL\
                                          4C 02729

                                          OFFC 00000 SCAN_INDEX:
                                                     .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11          ; 2617
                                 5E       90   AE  9E 00002         MOVAB   -112(SP), SP
                          00000000'       EF   DD  00006         PUSHL   VOLUME_COUNT                      ; 2645
                                          58   D4  0000C         CLRL    RVN
                                      08B7    30  0000E         BSBW    96$
    0040   8F        00                 6E       00  2C 00011 1$:      MOVC5   #0, (SP), #0, #64, FIB        ; 2654
                          00000000'       EF       00018
                                 50 00000000'FF48  DE  0001D         MOVAL   @ACCTL[RVN], R0                   ; 2655
                    00000000'  EF         FC   A0  D0  00025         MOVL    -4(R0), FIB
                    00000000'  EF 00010001     8F  D0  0002D         MOVL    #65537, FIB+4                     ; 2656
                    00000000'  EF              58  B0  00038         MOVW    RVN, FIB+8                        ; 2658
                                               7E  7C  0003F         CLRQ    -(SP)                             ; 2663
                                               7E  7C  00041         CLRQ    -(SP)
                                               7E  D4  00043         CLRL    -(SP)
                                          D8BD     CF  9F  00045         PUSHAB  FIB_DESC
```

```
                                7E  7C 00049       CLRQ    -(SP)                        ;
                     00000000'  EF  9F 0004B       PUSHAB  IOSB                         ;
              7E           72   8F  9A 00051       MOVZBL  #114, -(SP)                  ;
                     00000000'  EF  DD 00055       PUSHL   CHANNEL                      ;
                                7E  D4 0005B       CLRL    -(SP)                        ;
         00000000G  00          0C  FB 0005D       CALLS   #12, SYS$QIOW                ;
                    6E          50  D0 00064       MOVL    R0, STATUS                   ;
                    0A          6E  E9 00067       BLBC    STATUS, 2$                   ; 2664
                    6E 00000000' EF  3C 0006A      MOVZWL  IOSB, STATUS                 ;
                    13          6E  E8 00071       BLBS    STATUS, 3$                   ; 2665
                                6E  DD 00074 2$:   PUSHL   STATUS                       ;
                                58  DD 00076       PUSHL   RVN                          ;
                                01  DD 00078       PUSHL   #1                           ;
                     00000000G  8F  DD 0007A       PUSHL   #VERIFY$_OPENINDEX           ;
         00000000G  00          04  FB 00080       CALLS   #4, LIB$SIGNAL               ;
                    50 00000000'FF48 DE 00087 3$:  MOVAL   @HEADER_OFFSET[RVN], R0      ; 2671
   08   AE      FC  A0          01  C1 0008F       ADDL3   #1, -4(R0), VBN              ;
                    50 00000000'FF48 DE 00095 4$:  MOVAL   @EOF[RVN], R0                ; 2672
                FC  A0      08  AE  D1 0009D       CMPL    VBN, -4(R0)                  ;
                                03  1B 000A2       BLEQU   5$                           ;
                              0807  31 000A4       BRW     95$                          ;
        50      FC  A0      08  AE  C3 000A7 5$:   SUBL3   VBN, -4(R0), R0              ; 2681
                                50  D6 000AD       INCL    R0                           ;
         00000040   8F          50  D1 000AF       CMPL    R0, #64                      ;
                                04  1B 000B6       BLEQU   6$                           ;
                    50      40  8F  9A 000B8       MOVZBL  #64, R0                      ;
                14  AE          50  D0 000BC 6$:   MOVL    R0, READ_COUNT               ;
                                7E  7C 000C0       CLRQ    -(SP)                        ; 2688
                                7E  D4 000C2       CLRL    -(SP)                        ;
                            14  AE  DD 000C4       PUSHL   VBN                          ;
        7E      24  AE          09  78 000C7       ASHL    #9, READ_COUNT, -(SP)        ;
                     00000000'  EF  9F 000CC       PUSHAB  BUFFER                       ;
                                7E  7C 000D2       CLRQ    -(SP)                        ;
                     00000000'  EF  9F 000D4       PUSHAB  IOSB                         ;
                                31  DD 000DA       PUSHL   #49                          ;
                     00000000'  EF  DD 000DC       PUSHL   CHANNEL                      ;
                                7E  D4 000E2       CLRL    -(SP)                        ;
         00000000G  00          0C  FB 000E4       CALLS   #12, SYS$QIOW                ;
                    6E          50  D0 000EB       MOVL    R0, STATUS                   ;
                    0D          6E  E9 000EE       BLBC    STATUS, 7$                   ; 2689
                    6E 00000000' EF  3C 000F1      MOVZWL  IOSB, STATUS                 ;
                    03          6E  E9 000F8       BLBC    STATUS, 7$                   ; 2696
                              00C3  31 000FB       BRW     15$                          ;
                    53          01  CE 000FE 7$:   MNEGL   #1, XVBN                     ; 2699
                              00B3  31 00101       BRW     13$                          ;
        50          53          09  78 00104 8$:   ASHL    #9, XVBN, R0                 ; 2709
                    54 00000000'EF40 9E 00108      MOVAB   BUFFER[R0], HEADER           ;
                                7E  7C 00110       CLRQ    -(SP)                        ; 2716
                                7E  D4 00112       CLRL    -(SP)                        ;
        52          53      14  AE  C1 00114       ADDL3   VBN, XVBN, R2                ;
                                52  DD 00119       PUSHL   R2                           ;
                    7E    0200  8F  3C 0011B       MOVZWL  #512, -(SP)                  ;
                                54  DD 00120       PUSHL   HEADER                       ;
                                7E  7C 00122       CLRQ    -(SP)                        ;
                     00000000'  EF  9F 00124       PUSHAB  IOSB                         ;
                                31  DD 0012A       PUSHL   #49                          ;
                     00000000'  EF  DD 0012C       PUSHL   CHANNEL                      ;
```

```
                                    7E  D4 00132          CLRL    -(SP)                                                   ;
                00000000G     00    0C  FB 00134          CALLS   #12, SYS$QIOW                                           ;
                              6E    50  D0 0013B          MOVL    R0, STATUS                                              ;
                              0A    6E  E9 0013E          BLBC    STATUS, 9$                                              ; 2717
                              6E 00000000'  EF  3C 00141  MOVZWL  IOSB, STATUS                                            ;
                              6C    6E  E8 00148          BLBS    STATUS, 13$                                             ; 2718
                              50 00000000'FF48  DE 0014B  9$:     MOVAL   @HEADER_OFFSET[RVN], R0                         ; 2724
                              52      FC  A0  C2 00153    SUBL2   -4(R0), FILE_NUMBER                                     ;
                        70    AE        52  B0 00157      MOVW    FILE_NUMBER, FILE_ID                                    ; 2725
          50      52    08              10  EF 0015B      EXTZV   #16, #8, FILE_NUMBER, R0                                ; 2726
                        75    AE        50  90 00160      MOVB    R0, FILE_ID+5                                           ;
                              02 00000000'  EF  D1 00164  CMPL    STRUCTURE_LEVEL, #2                                     ; 2727
                                    07  12 0016B          BNEQ    10$                                                     ;
                        72    AE    0A  A4  B0 0016D      MOVW    10(HEADER), FILE_ID+2                                   ; 2728
                                    05  11 00172          BRB     11$                                                     ;
                        72    AE    04  A4  B0 00174  10$:  MOVW    4(HEADER), FILE_ID+2                                ; 2729
                        74    AE        58  90 00179  11$:  MOVB    RVN, FILE_ID+4                                      ; 2730
                                    6E  DD 0017D          PUSHL   STATUS                                                  ; 2737
                                    54  DD 0017F          PUSHL   HEADER                                                  ; 2736
                              78    AE  9F 00181          PUSHAB  FILE_ID                                                 ; 2735
                        00000000G     8F  DD 00184        PUSHL   #VERIFY$_READHEADER                                     ;
                     0000V  CF        04  FB 0018A        CALLS   #4, HEADER_ERROR                                        ;
                              07    A4  94 0018F          CLRB    7(HEADER)                                               ; 2744
                                    7E  D4 00192          CLRL    -(SP)                                                   ; 2745
                     0000V  CF        01  FB 00194        CALLS   #1, DO_REPAIR                                           ;
                              0C      50  E9 00199        BLBC    R0, 12$                                                 ;
                                    54  DD 0019C          PUSHL   HEADER                                                  ; 2747
                              74    AE  9F 0019E          PUSHAB  FILE_ID                                                 ;
                     0000V  CF        02  FB 001A1        CALLS   #2, DELETE_HEADER                                       ;
                                    0F  11 001A6          BRB     13$                                                     ;
                              50 00000000'FF48  DE 001A8  12$:    MOVAL   @IMAP[RVN], R0                                  ; 2749
                                    52  D7 001B0          DECL    R2                                                      ;
                00      FC    B0        52  E5 001B2      BBCC    R2, @-4(R0), 13$                                        ;
                02            53    14  AE  F2 001B7  13$:  AOBLSS  READ_COUNT, XVBN, 14$                               ; 2699
                                    03  11 001BC          BRB     15$                                                     ;
                                FF43    31 001BE  14$:  BRW     8$                                                      ;
                              56 00000000'  EF  9E 001C1  15$:  MOVAB   BUFFER, HEADER                                  ; 2758
                        10    AE        01  CE 001C8      MNEGL   #1, XVBN                                                ; 2759
                                06C9    31 001CC          BRW     92$                                                     ;
                              50    08  AE  D0 001CF  16$:  MOVL    VBN, R0                                             ; 2771
                51            50    10  AE  C1 001D3      ADDL3   XVBN, R0, R1                                            ;
                              50 00000000'FF48  DE 001D8    MOVAL   @HEADER_OFFSET[RVN], R0                               ;
                57            51      FC  A0  C3 001E0    SUBL3   -4(R0), R1, FILE_NUMBER                                 ;
                        70    AE        57  B0 001E5      MOVW    FILE_NUMBER, FILE_ID                                    ; 2772
          50    57      08              10  EF 001E9      EXTZV   #16, #8, FILE_NUMBER, R0                                ; 2773
                        75    AE        50  90 001EE      MOVB    R0, FILE_ID+5                                           ;
                              02 00000000'  EF  D1 001F2  CMPL    STRUCTURE_LEVEL, #2                                     ; 2774
                                    07  12 001F9          BNEQ    17$                                                     ;
                        72    AE    0A  A6  B0 001FB      MOVW    10(HEADER), FILE_ID+2                                   ; 2775
                                    05  11 00200          BRB     18$                                                     ;
                        72    AE    04  A6  B0 00202  17$:  MOVW    4(HEADER), FILE_ID+2                                ; 2776
                        74    AE        58  90 00207  18$:  MOVB    RVN, FILE_ID+4                                      ; 2777
                                    70  AE  9F 0020B          PUSHAB  FILE_ID                                                 ; 2785
                                    56  DD 0020E          PUSHL   HEADER                                                  ;
                     0000V  CF        02  FB 00210        CALLS   #2, VERIFY_HEADER                                       ;
                              6E      50  D0 00215        MOVL    R0, STATUS                                              ;
                                    01  00000000'  EF  D1 00218  CMPL    STRUCTURE_LEVEL, #1                             ; 2787
```

```
                                          13  12 0021F        BNEQ    19$                                            ;
                         50 00000000'FF48  DE 00221           MOVAL   @[MAP[RVN], R0                                ; 2788
                         54          FF A7  9E 00229          MOVAB   -1(R7), R4                                     ;
           02        FC  B0             54  E0 0022D          BBS     R4, @-4(R0), 19$                               ;
                                        6E  D4 00232          CLRL    STATUS                                         ; 2790
                         50 00000000'  EF  D2 00234  19$:     MCOML   DUAL_ALLOC_PASS, R0                            ; 2803
                         03             6E  E8 0023B          BLBS    STATUS, 20$                                    ; 2793
                                      061E  31 0023E          BRW     90$                                            ;
                         5B             66  9A 00241  20$:    MOVZBL  (HEADER), R11                                  ; 2799
                         59           664B  3E 00244          MOVAW   (HEADER)[R11], IDENT_AREA                      ;
                         5A          01 A6  9A 00248          MOVZBL  1(HEADER), R10                                 ; 2800
           0C            AE           664A  3E 0024C          MOVAW   (HEADER)[R10], MAP_AREA                        ;
                         03             50  E8 00251          BLBS    R0, 21$                                        ; 2803
                                      00E8  31 00254          BRW     30$                                            ;
           03 00000000'  EF             01  E0 00257  21$:    BBS     #1, QUAL, 22$                                  ; 2806
                                      0092  31 0025F          BRW     28$                                            ;
           02 00000000'  EF             D1 00262  22$:        CMPL    STRUCTURE_LEVEL, #2                            ; 2816
                                        3D  12 00269          BNEQ    25$                                            ;
  0057  8F       20      69             14  2C 0026B          MOVC5   #20, (IDENT_AREA), #32, #87, FILENAME          ; 2820
                                   18   AE     00272                                                                 ;
                         5A             5B  C2 00274          SUBL2   R11, R10                                       ; 2824
                         5A             02  C4 00277          MULL2   #2, R10                                        ;
              00000078   8F             5A  D1 0027A          CMPL    R10, #120                                      ; 2825
                                        08  1F 00281          BLSSU   23$                                            ;
      2C  AE       36    A9      0042   8F  28 00283          MOVC3   #66, 54(IDENT_AREA), FILENAME+20              ; 2830
      18  AE     0057    8F             20  3A 0028B  23$:    LOCC    #32, #87, FILENAME                             ; 2833
                                        02  12 00292          BNEQ    24$                                            ;
                                        51  D4 00294          CLRL    R1                                             ;
                         50          18 AE  9E 00296  24$:    MOVAB   FILENAME, R0                                   ; 2834
           50            51             50  C3 0029A          SUBL3   R0, R1, LENGTH                                 ;
                         51          3C A6  D0 0029E          MOVL    60(HEADER), OWNER                              ; 2836
                         52          04 A6  3C 002A2          MOVZWL  4(HEADER), EXTENSION_SEGMENT                   ; 2837
                                        1E  11 002A6          BRB     26$                                            ; 2816
                                     18 AE  9F 002A8  25$:    PUSHAB  FILENAME                                       ; 2841
                                     FA A9  9F 002AB          PUSHAB  -6(IDENT_AREA)                                 ; 2842
              00000000G  EF             02  FB 002AE          CALLS   #2, MAKE_STRING                                ;
                         51          08 A6  9A 002B5          MOVZBL  8(HEADER), OWNER                               ; 2845
                         52          09 A6  9A 002B9          MOVZBL  9(HEADER), R2                                  ; 2846
        51       10      10             52  F0 002BD          INSV    R2, #16, #16, OWNER                            ;
                         52          0C BE  9A 002C2          MOVZBL  @MAP_AREA, EXTENSION_SEGMENT                   ; 2847
                                        51  DD 002C6  26$:    PUSHL   OWNER                                          ; 2856
                                     1C AE  9F 002C8          PUSHAB  FILENAME                                       ;
                                        50  DD 002CB          PUSHL   LENGTH                                         ;
                         7E          FC AD  9A 002CD          MOVZBL  FILE_ID+4, -(SP)                               ;
                         7E          FA AD  3C 002D1          MOVZWL  FILE_ID+2, -(SP)                               ;
                                        57  DD 002D5          PUSHL   FILE_NUMBER                                    ;
                                   FCF8 CF  9F 002D7          PUSHAB  P.ABK                                          ;
                  0000V  CF             07  FB 002DB          CALLS   #7, FAO                                        ;
                                        52  D5 002E0          TSTL    EXTENSION_SEGMENT                              ; 2858
                                        08  13 002E2          BEQL    27$                                            ;
                                        52  DD 002E4          PUSHL   EXTENSION_SEGMENT                              ; 2860
                                   FD06 CF  9F 002E6          PUSHAB  P.ABL                                          ;
                  0000V  CF             02  FB 002EA          CALLS   #2, FAO                                        ;
                  0000V  CF             00  FB 002EF  27$:    CALLS   #0, EOL                                        ; 2862
                         50 00000000'FF48  DE 002F4  28$:     MOVAL   @[MAP[RVN], R0                                ; 2870
                         54          FF A7  9E 002FC          MOVAB   -1(R7), R4                                     ;
           00        FC  B0             54  E2 00300          BBSS    R4, @-4(R0), 29$                               ;
```

```
                        50 00000000'FF48   DE 00305 29$:    MOVAL   @SEQMAP[RVN], R0                        ; 2871
                    FC B044        72   AE B0 00309         MOVW    FILE_ID+2, @-4(R0)[R4]
                        02 00000000'    EF D1 00313         CMPL    STRUCTURE_LEVEL, #2                     ; 2872
                                    23   12 00319         BNEQ    30$
                        51 00000000'FF48   DE 0031C         MOVAL   @BACKMAP[RVN], R1                       ; 2878
                50      54             03   C5 00324         MULL3   #3, R4, R0
                        50          FC B140   3E 00328         MOVAW   @-4(R1)[R0], BACK_ID
                        60          42   A6 D0 0032D         MOVL    66(HEADER), (BACK_ID)                   ; 2879
                    04  A0          46   A6 B0 00331         MOVW    70(HEADER), 4(BACK_ID)                  ; 2881
                        04          A0   95 00336         TSTB    4(BACK_ID)                              ; 2882
                                    04   12 00339         BNEQ    30$
                    04  A0                58 90 0033B         MOVB    RVN, 4(BACK_ID)
                                    51   D4 0033F 30$:      CLRL    R1                                      ; 2891
                        02 00000000'    EF D1 00341         CMPL    STRUCTURE_LEVEL, #2
                                    07   12 00348         BNEQ    31$
                                    51   D6 0034A         INCL    R1
                                04  A6   B5 0034C         TSTW    4(HEADER)                               ; 2892
                                    03   11 0034F         BRB     32$
                                0C  BE   95 00351 31$:      TSTB    @MAP_AREA                               ; 2893
                                    03   13 00354 32$:      BEQL    33$
                                  04F3   31 00356         BRW     89$
                        03 00000000'    EF E9 00359 33$:      BLBC    DUAL_ALLOC_PASS, 34$                    ; 2902
                                  0138   31 00360         BRW     45$
                        50 00000000'FF48   DE 00363 34$:      MOVAL   @LOSTMAP[RVN], R0                       ; 2908
                        54          FF   A7 9E 00368         MOVAB   -1(R7), R4
                00      FC          B0   54 E2 0036F         BBSS    R4, @-4(R0), 35$
                            0A          51   E9 00374 35$:      BLBC    R1, 36$                                 ; 2913
                            53      34  A6   9E 00377         MOVAB   52(R6), FCH                             ; 2916
                            52      14  A6   9E 0037B         MOVAB   20(R6), FAT                             ; 2917
                                        08   11 0037F         BRB     37$                                     ; 2913
                            53      0C  A6   9E 00381 36$:      MOVAB   12(R6), FCH                             ; 2921
                            52      0E  A6   9E 00385         MOVAB   14(R6), FAT                             ; 2922
                                        63   B5 00389 37$:      TSTW    (FCH)                                   ; 2928
                                        2F   18 0038B         BGEQ    38$
                                        56   DD 0038D         PUSHL   HEADER                                  ; 2931
                                    74  AE   9F 0038F         PUSHAB  FILE_ID
                                00000000G   8F   DD 00392         PUSHL   #VERIFY$_DELHEADER
                        0000V   CF          03   FB 00398         CALLS   #3, HEADER_ERROR
                        0000V   CF          00   FB 0039D         CALLS   #0, DO_REPAIR                   ; 2932
                                17          50   E9 003A2         BLBC    R0, 38$
                                        70  AE   9F 003A5         PUSHAB  FILE_ID                         ; 2935
                                        03   DD 003A8         PUSHL   #3
                        0000V   CF          02   FB 003AA         CALLS   #2, ENTER_WORK
                                50 00000000'FF48   DE 003AF         MOVAL   @SEQMAP[RVN], R0                ; 2936
                            FC B044         01   AE 003B7         MNEGW   #1, @-4(R0)[R4]
                48              63          06   E1 003BC 38$:      BBC     #6, (FCH), 40$                  ; 2943
                                        56   DD 003C0         PUSHL   HEADER                          ; 2946
                                    74  AE   9F 003C2         PUSHAB  FILE_ID
                                00000000G   8F   DD 003C5         PUSHL   #VERIFY$_LOCKHEADER
                        0000V   CF          03   FB 003CB         CALLS   #3, HEADER_ERROR
                                        03   DD 003D0         PUSHL   #3                              ; 2947
                        0000V   CF          01   FB 003D2         CALLS   #1, DO_REPAIR
                                6E          50   D0 003D7         MOVL    R0, STATUS
                                2B          6E   E9 003DA         BLBC    STATUS, 40$
                19              6E          01   E1 003DD         BBC     #1, STATUS, 39$                 ; 2950
                                        70  AE   9F 003E1         PUSHAB  FILE_ID                         ; 2953
                                        03   DD 003E4         PUSHL   #3
```

```
                0000V  CF                    02  FB 003E6           CALLS   #2, ENTER_WORK                                  ;
                    50 00000000'FF48         48  DE 003EB           MOVAL   @SEQMAP[RVN], R0                            2954
                  FC B044                    01  AE 003F3           MNEGW   #1, @-4(R0)[R4]                             ;
                                             0E  11 003F8           BRB     40$                                         2950
                       63          40        8F  BA 003FA 39$:      BICB2   #64, (FCH)                                  2958
                                             56  DD 003FE           PUSHL   HEADER                                      2959
                                74           AE  9F 00400           PUSHAB  FILE_ID                                     ;
                0000V  CF                    02  FB 00403           CALLS   #2, WRITE_HEADER                            ;
                    10  63                   0E  E1 00408 40$:      BBC     #14, (FCH), 41$                             2967
                                             56  DD 0040C           PUSHL   HEADER                                      2969
                                74           AE  9F 0040E           PUSHAB  FILE_ID                                     ;
                                00000000G    8F  DD 00411           PUSHL   #VERIFY$_BBLHEADER                          ;
                0000V  CF                    03  FB 00417           CALLS   #3, HEADER_ERROR                            ;
                       04                    57  D1 0041C 41$:      CMPL    FILE_NUMBER, #4                             2975
                                             30  13 0041F           BEQL    44$                                         ;
                       02 00000000'          EF  D1 00421           CMPL    STRUCTURE_LEVEL, #2                         2978
                                             07  12 00428           BNEQ    42$                                         ;
                    22    35     A6          05  E1 0042A           BBC     #5, 53(HEADER), 44$                         2980
                                             13  11 0042F           BRB     43$                                         ;
                       01                    62  91 00431 42$:      CMPB    (FAT), #1                                   2982
                                             1B  12 00434           BNEQ    44$                                         ;
                       10          02        A2  B1 00436           CMPW    2(FAT), #16                                 2983
                                             15  12 0043A           BNEQ    44$                                         ;
                  1A7A 8F          06        A9  B1 0043C           CMPW    6(IDENT_AREA), #6778                        2984
                                             0D  12 00442           BNEQ    44$                                         ;
                       50 00000000'FF48      48  DE 00444 43$:      MOVAL   @DIRMAP[RVN], R0                            2987
                    00    FC     B0          54  E2 0044C           BBSS    R4, @-4(R0), 44$                            ;
                                             56  DD 00451 44$:      PUSHL   HEADER                                      2997
                                74           AE  9F 00453           PUSHAB  FILE_ID                                     2994
                                000000000G   8F  DD 00456           PUSHL   #VERIFY$_FUTCREDAT                          ;
                                      19     A9  9F 0045C           PUSHAB  25(IDENT_AREA)                              ;
                                      16     A9  9F 0045F           PUSHAB  22(IDENT_AREA)                              2993
                0000V  CF                    05  FB 00462           CALLS   #5, CHECK_DATE                              2994
                                             56  DD 00467           PUSHL   HEADER                                      3007
                                74           AE  9F 00469           PUSHAB  FILE_ID                                     3004
                                000000000G   8F  DD 0046C           PUSHL   #VERIFY$_FUTREVDAT                          ;
                                      0C     A9  9F 00472           PUSHAB  12(IDENT_AREA)                              ;
                                      1E     A9  9F 00475           PUSHAB  30(IDENT_AREA)                              3003
                0000V  CF                    05  FB 00478           CALLS   #5, CHECK_DATE                              3004
                       02 00000000'          EF  D1 0047D           CMPL    STRUCTURE_LEVEL, #2                         3012
                                             15  12 00484           BNEQ    45$                                         ;
                                             56  DD 00486           PUSHL   HEADER                                      3019
                                74           AE  9F 00488           PUSHAB  FILE_ID                                     3015
                                000000000G   8F  DD 0048B           PUSHL   #VERIFY$_FUTBAKDAT                          ;
                                             7E  D4 00491           CLRL    -(SP)                                       ;
                                      2E     A9  9F 00493           PUSHAB  46(IDENT_AREA)                              ;
                0000V  CF                    05  FB 00496           CALLS   #5, CHECK_DATE                              ;
                                00000000'    EF  D4 0049B 45$:      CLRL    TOTAL_SIZE                                  3025
                                             01  DD 004A1           PUSHL   #1                                          3026
                                74           AE  9F 004A3           PUSHAB  FILE_ID                                     ;
                                             56  DD 004A6           PUSHL   HEADER                                      ;
                                      0140   8F  BB 004A8           PUSHR   #^M<R6,R8>                                  ;
                0000V  CF                    05  FB 004AC           CALLS   #5, MAP_PROCESS                             ;
                                             5B  D4 004B1           CLRL    EXT_SEQ                                     3032
                       02 00000000'          EF  D1 004B3           CMPL    STRUCTURE_LEVEL, #2                         3035
                                             05  12 004BA           BNEQ    46$                                         ;
                                      0E     A6  B5 004BC           TSTW    14(HEADER)                                  3036
```

```
                                  07  11 004BF       BRB     47$                                      ;
             50       0C  AE          02  C1 004C1 46$: ADDL3   #2, MAP_AREA, R0                         ; 3037
                                      60  B5 004C6       TSTW    (R0)                                     ;
                                      6A  13 004C8 47$: BEQL    54$                                      ;
                          54          56  D0 004CA       MOVL    HEADER, EXT_HDR                          ; 3048
                          5A     0C   AE  D0 004CD       MOVL    MAP_AREA, EXT_MAP_AREA                   ; 3049
                          55          58  D0 004D1       MOVL    RVN, EXT_RVN                             ; 3050
                      68  AE     70   AE  D0 004D4       MOVL    FILE_ID, PREV_FILE_ID                    ; 3051
                      6C  AE     74   AE  B0 004D9       MOVW    FILE_ID+4, PREV_FILE_ID+4                ; 3053
                                      50  D4 004DE 48$: CLRL    R0                                       ; 3056
                      02 00000000'    EF  D1 004E0       CMPL    STRUCTURE_LEVEL, #2                      ;
                                      07  12 004E7       BNEQ    49$                                      ;
                                      50  D6 004E9       INCL    R0                                       ;
                                 0E   A4  B5 004EB       TSTW    14(EXT_HDR)                              ; 3057
                                      03  11 004EE       BRB     50$                                      ;
                                 02   AA  B5 004F0 49$: TSTW    2(EXT_MAP_AREA)                          ; 3058
                                      3F  13 004F3 50$: BEQL    54$                                      ;
                          16          50  E9 004F5       BLBC    R0, 51$                                  ; 3069
                          53     0E   A4  3C 004F8       MOVZWL  14(EXT_HDR), EXT_FILE_NUMBER             ; 3072
     53          08       10     13   A4  F0 004FC       INSV    19(EXT_HDR), #16, #8, EXT_FILE_NUMBER    ; 3073
                      60  AE     0E   A4  D0 00502       MOVL    14(EXT_HDR), EXT_FILE_ID                 ; 3074
                      64  AE     12   A4  B0 00507       MOVW    18(EXT_HDR), EXT_FILE_ID+4               ; 3076
                                      0D  11 0050C       BRB     52$                                      ; 3069
                          53     02   AA  3C 0050E 51$: MOVZWL  2(EXT_MAP_AREA), EXT_FILE_NUMBER         ; 3080
                      60  AE     02   AA  D0 00512       MOVL    2(EXT_MAP_AREA), EXT_FILE_ID             ; 3081
                      64  AE          01  B0 00517       MOVW    #1, EXT_FILE_ID+4                        ; 3083
                                 64   AE  95 0051B 52$: TSTB    EXT_FILE_ID+4                            ; 3085
                                      04  12 0051E       BNEQ    53$                                      ;
                      64  AE          55  90 00520       MOVB    EXT_RVN, EXT_FILE_ID+4                   ;
                          55     64   AE  9A 00524 53$: MOVZBL  EXT_FILE_ID+4, EXT_RVN                   ; 3086
                          0E          50  E9 00528       BLBC    R0, 56$                                  ; 3094
                          07          53  D1 0052B       CMPL    EXT_FILE_NUMBER, #7                      ; 3095
                                      09  12 0052E       BNEQ    56$                                      ;
                      07         62   AE  B1 00530       CMPW    EXT_FILE_ID+2, #7                        ; 3096
                                      03  12 00534 54$: BNEQ    56$                                      ;
                                    01C0  31 00536 55$: BRW     79$                                      ;
              00000000'   EF          55  D1 00539 56$: CMPL    EXT_RVN, VOLUME_COUNT                    ; 3106
                                      12  1A 00540       BGTRU   57$                                      ;
                          51     FF   A3  9E 00542       MOVAB   -1(R3), R1                               ; 3110
                          50 00000000'FF45 DE 00546      MOVAL   @MAXFILIDX[EXT_RVN], R0                  ;
                      FC  A0          51  D1 0054E       CMPL    R1, -4(R0)                               ;
                                      26  1B 00552       BLEQU   58$                                      ;
                      62 00000000'    EF  E8 00554 57$: BLBS    DUAL_ALLOC_PASS, 62$                     ; 3114
                                      56  DD 0055B       PUSHL   HEADER                                   ; 3117
                                 74   AE  9F 0055D       PUSHAB  FILE_ID                                  ;
                         00000000G    8F  DD 00560       PUSHL   #VERIFY$_INVEXTFID                       ;
                 0000V    CF          03  FB 00566       CALLS   #3, HEADER_ERROR                         ;
                 0000V    CF          00  FB 0056B       CALLS   #0, DO_REPAIR                            ; 3118
                          C3          50  E9 00570       BLBC    R0, 55$                                  ;
                                      56  DD 00573       PUSHL   HEADER                                   ; 3120
                                 74   AE  9F 00575       PUSHAB  FILE_ID                                  ;
                                      7F  11 00578       BRB     65$                                      ;
                         00000000'    EF  9F 0057A 58$: PUSHAB  BUFFER_2                                 ; 3129
                                 64   AE  9F 00580       PUSHAB  EXT_FILE_ID                              ;
                 0000V    CF          02  FB 00583       CALLS   #2, READ_HEADER                          ;
                          09          50  E8 00588       BLBS    R0, 59$                                  ;
                      2B 00000000'    EF  E8 0058B       BLBS    DUAL_ALLOC_PASS, 62$                     ; 3132
```

```
                                          43  11 00592       BRB     64$                                           ; 3135
                      54 00000000'        EF  9E 00594 59$:  MOVAB   BUFFER_2, EXT_HDR                             ; 3147
                      50       01         A4  9A 0059B       MOVZBL  1(EXT_HDR), R0                                ; 3148
                      5A               6440   3E 0059F       MOVAW   (EXT_HDR)[R0], EXT_MAP_AREA
                                          5B  D6 005A3       INCL    EXT_SEQ                                       ; 3149
                      02 00000000'        EF  D1 005A5       CMPL    STRUCTURE_LEVEL, #2                           ; 3158
                                          08  12 005AC       BNEQ    60$
  5B        04   A4           10          00  ED 005AE       CMPZV   #0, #16, 4(EXT_HDR), EXT_SEQ                  ; 3159
                                          05  11 005B4       BRB     61$
  5B             6A           08          00  ED 005B6 60$:  CMPZV   #0, #8, (EXT_MAP_AREA), EXT_SEQ               ; 3160
                                          44  13 005BB 61$:  BEQL    67$
                      03 00000000'        EF  E9 005BD 62$:  BLBC    DUAL_ALLOC_PASS, 63$                          ; 3164
                                        02CC  31 005C4       BRW     91$
                                          54  DD 005C7 63$:  PUSHL   EXT_HDR                                       ; 3167
                                64        AE  9F 005C9       PUSHAB  EXT_FILE_ID
                         00000000G        8F  DD 005CC       PUSHL   #VERIFY$_INVEXTHDR
                 0000V   CF               03  FB 005D2       CALLS   #3, HEADER_ERROR
                 0000V   CF               00  FB 005D7 64$:  CALLS   #0, DO_REPAIR                                 ; 3168
                         1F               50  E9 005DC       BLBC    R0, 66$
                         00000000'        EF  9F 005DF       PUSHAB  BUFFER_2                                      ; 3171
                                6C        AE  9F 005E5       PUSHAB  PREV_FILE_ID
                 0000V   CF               02  FB 005E8       CALLS   #2, READ_HEADER
                         0E               50  E9 005ED       BLBC    R0, 66$
                         00000000'        EF  9F 005F0       PUSHAB  BUFFER_2                                      ; 3173
                                6C        AE  9F 005F6       PUSHAB  PREV_FILE_ID
                 0000V   CF               02  FB 005F9 65$:  CALLS   #2, CLEAR_EXT_FID
                                        00F8  31 005FE 66$:  BRW     79$                                           ; 3163
                    6B   AE     60        AE  D0 00601 67$:  MOVL    EXT_FILE_ID, PREV_FILE_ID                     ; 3183
                    6C   AE     64        AE  B0 00606       MOVW    EXT_FILE_ID+4, PREV_FILE_ID+4                 ; 3185
                                          7E  D4 0060B       CLRL    -(SP)                                         ; 3190
                                74        AE  9F 0060D       PUSHAB  FILE_ID
                              0070        8F  BB 00610       PUSHR   #^M<R4,R5,R6>
                 0000V   CF               05  FB 00614       CALLS   #5, MAP_PROCESS
                      03 00000000'        EF  E9 00619       BLBC    DUAL_ALLOC_PASS, 68$                          ; 3193
                                        FEBB  31 00620       BRW     48$
                      02 00000000'        EF  D1 00623 68$:  CMPL    STRUCTURE_LEVEL, #2                           ; 3200
                                          5D  12 0062A       BNEQ    72$
                      50        46        A4  9A 0062C       MOVZBL  70(EXT_HDR), BCK_RVN                          ; 3206
                                          04  12 00630       BNEQ    69$                                           ; 3207
                      50        64        AE  9A 00632       MOVZBL  EXT_FILE_ID+4, BCK_RVN
                    70   AE     42        A4  B1 00636 69$:  CMPW    66(EXT_HDR), FILE_ID                          ; 3209
                                          16  12 0063B       BNEQ    70$
                    72   AE     44        A4  B1 0063D       CMPW    68(EXT_HDR), FILE_ID+2                        ; 3210
                                          0F  12 00642       BNEQ    70$
                    75   AE     47        A4  91 00644       CMPB    71(EXT_HDR), FILE_ID+5                        ; 3211
                                          08  12 00649       BNEQ    70$
  50        74   AE           08          00  ED 0064B       CMPZV   #0, #8, FILE_ID+4, BCK_RVN                    ; 3212
                                          36  13 00651       BEQL    72$
                                          54  DD 00653 70$:  PUSHL   EXT_HDR                                       ; 3216
                                64        AE  9F 00655       PUSHAB  EXT_FILE_ID                                   ; 3215
                         00000000G        8F  DD 00658       PUSHL   #VERIFY$_INVEXTBACK
                 0000V   CF               03  FB 0065E       CALLS   #3, HEADER_ERROR
                 0000V   CF               00  FB 00663       CALLS   #0, DO_REPAIR                                 ; 3217
                         1E               50  E9 00668       BLBC    R0, 72$
                    42   A4     70        AE  D0 0066B       MOVL    FILE_ID, 66(EXT_HDR)                          ; 3220
                    46   A4     74        AE  B0 00670       MOVW    FILE_ID+4, 70(EXT_HDR)                        ; 3222
                    64   AE     46        A4  91 00675       CMPB    70(EXT_HDR), EXT_FILE_ID+4                    ; 3223
```

```
                                  03   12 0067A        BNEQ    71$                                 ;
                         46       A4   94 0067C        CLRB    70(EXT_HDR)                         ; 3224
                                  54   DD 0067F 71$:   PUSHL   EXT_HDR                             ; 3225
                         64       AE   9F 00681        PUSHAB  EXT_FILE_ID                         ;
            0000V  CF             02   FB 00684        CALLS   #2, WRITE_HEADER                    ;
                   50 00000000'FF45    DE 00689 72$:   MOVAL   @EXTMAP[EXT_RVN], R0                ; 3234
                                  53   D7 00691        DECL    R3                                  ;
        10         FC  B0         53   E3 00693        BBCS    R3, @-4(R0), 73$                    ;
                                  54   DD 00698        PUSHL   EXT_HDR                             ; 3237
                         64       AE   9F 0069A        PUSHAB  EXT_FILE_ID                         ; 3236
                    00000000G     8F   DD 0069D        PUSHL   #VERIFY$_MULTEXTHDR                 ;
            0000V  CF             03   FB 006A3        CALLS   #3, HEADER_ERROR                    ;
                   02 00000000'   EF   D1 006A8 73$:   CMPL    STRUCTURE_LEVEL, #2                 ; 3245
                                  07   12 006AF        BNEQ    74$                                 ;
              3C   A6        3C   A4   D1 006B1        CMPL    60(EXT_HDR), 60(HEADER)             ; 3246
                                  05   11 006B6        BRB     75$                                 ;
              08   A6        08   A4   B1 006B8 74$:   CMPW    8(EXT_HDR), 8(HEADER)               ; 3247
                                  37   13 006BD 75$:   BEQL    78$                                 ;
                                  54   DD 006BF        PUSHL   EXT_HDR                             ; 3251
                         64       AE   9F 006C1        PUSHAB  EXT_FILE_ID                         ;
                    00000000G     8F   DD 006C4        PUSHL   #VERIFY$_WRONGOWNER                 ;
            0000V  CF             03   FB 006CA        CALLS   #3, HEADER_ERROR                    ;
            0000V  CF             00   FB 006CF        CALLS   #0, DO_REPAIR                       ; 3252
                   1F             50   E9 006D4        BLBC    R0, 78$                             ;
                   02 00000000'   EF   D1 006D7        CMPL    STRUCTURE_LEVEL, #2                 ; 3255
                                  07   12 006DE        BNEQ    76$                                 ;
              3C   A4        3C   A6   D0 006E0        MOVL    60(HEADER), 60(EXT_HDR)             ; 3256
                                  05   11 006E5        BRB     77$                                 ;
              08   A4        08   A6   B0 006E7 76$:   MOVW    8(HEADER), 8(EXT_HDR)               ; 3257
                                  54   DD 006EC 77$:   PUSHL   EXT_HDR                             ; 3258
                         64       AE   9F 006EE        PUSHAB  EXT_FILE_ID                         ;
            0000V  CF             02   FB 006F1        CALLS   #2, WRITE_HEADER                    ;
                                FDE5   31 006F6 78$:   BRW     48$                                 ; 3054
                   03 00000000'   EF   E9 006F9 79$:   BLBC    DUAL_ALLOC_PASS, 80$                ; 3266
                                0190   31 00700        BRW     91$                                 ;
        53         04  A2             10   9C 00703 80$:   ROTL    #16, 4(FAT), R3                 ; 3276
                   53 00000000'   EF   D1 00708        CMPL    TOTAL_SIZE, R3                      ;
                                  39   13 0070F        BEQL    82$                                 ;
                   01 00000000'   EF   D1 00711        CMPL    STRUCTURE_LEVEL, #1                 ; 3277
                                  05   12 00718        BNEQ    81$                                 ;
                   03             57   D1 0071A        CMPL    FILE_NUMBER, #3                     ;
                                  2B   1B 0071D        BLEQU   82$                                 ;
                                  56   DD 0071F 81$:   PUSHL   HEADER                              ; 3280
                         74       AE   9F 00721        PUSHAB  FILE_ID                             ;
                    00000000G     8F   DD 00724        PUSHL   #VERIFY$_BADHIBLK                   ;
            0000V  CF             03   FB 0072A        CALLS   #3, HEADER_ERROR                    ;
            0000V  CF             00   FB 0072F        CALLS   #0, DO_REPAIR                       ; 3281
                   13             50   E9 00734        BLBC    R0, 82$                             ;
     04    A2 00000000'   EF      10   9C 00737        ROTL    #16, TOTAL_SIZE, 4(FAT)             ; 3284
                                  56   DD 00740        PUSHL   HEADER                              ; 3285
                         74       AE   9F 00742        PUSHAB  FILE_ID                             ;
            0000V  CF             02   FB 00745        CALLS   #2, WRITE_HEADER                    ;
                         08       A2   D5 0074A 82$:   TSTL    8(FAT)                              ; 3291
                                  4C   13 0074D        BEQL    84$                                 ;
        53         08  A2             10   9C 0074F        ROTL    #16, 8(FAT), R3                 ; 3292
                                  50   D4 00754        CLRL    R0                                  ;
                         0C       A2   B5 00756        TSTW    12(FAT)                             ;
```

```
                                                02  12  00759          BNEQ    83$                                     ;
                                                50  D6  0075B          INCL    R0                                      ;
                                      53        50  C2  0075D 83$:     SUBL2   R0, R3                                  ;
                                00000000'  EF   53  D1  00760          CMPL    R3, TOTAL_SIZE                          ; 3293
                                                32  1B  00767          BLEQU   84$                                     ;
                                                56  DD  00769          PUSHL   HEADER                                  ; 3296
                                           74   AE  9F  0076B          PUSHAB  FILE_ID                                 ;
                                      00000000G 8F  DD  0076E          PUSHL   #VERIFY$_BADEFBLK                       ;
                                     0000V  CF      03  FB  00774          CALLS   #3, HEADER_ERROR                        ;
                                     0000V  CF      00  FB  00779          CALLS   #0, DO_REPAIR                           ; 3297
                                            1A      50  E9  0077E          BLBC    R0, 84$                                 ;
                                50 00000000'  EF      01  C1  00781          ADDL3   #1, TOTAL_SIZE, R0                      ; 3300
                             08 A2            50      10  9C  00789          ROTL    #16, R0, 8(FAT)                         ;
                                           0C   A2      B4  0078E          CLRW    12(FAT)                                 ; 3301
                                                56  DD  00791          PUSHL   HEADER                                  ; 3302
                                           74   AE  9F  00793          PUSHAB  FILE_ID                                 ;
                                     0000V  CF      02  FB  00796          CALLS   #2, WRITE_HEADER                        ;
                                0F 00000000'  EF      02  E1  0079B 84$:     BBC     #2, QUAL, 85$                           ; 3309
                                            04      57  D1  007A3          CMPL    FILE_NUMBER, #4                         ; 3311
                                                0A  1F  007A6          BLSSU   85$                                     ;
                                                56  DD  007A8          PUSHL   HEADER                                  ; 3313
                                           74   AE  9F  007AA          PUSHAB  FILE_ID                                 ;
                                     0000V  CF      02  FB  007AD          CALLS   #2, READ_CHECK                          ;
                                6F 00000000'  EF      04  E1  007B2 85$:     BBC     #4, QUAL, 88$                           ; 3318
                                50 00000000'FF48      DE  007BA          MOVAL   @OWNER[RVN], R0                         ; 3322
                                           54   FF      A7  9E  007C2          MOVAB   -1(R7), R4                              ;
                                           54          04  C4  007C6          MULL2   #4, R4                                  ;
                                02 00000000'          EF  D1  007C9          CMPL    STRUCTURE_LEVEL, #2                     ; 3321
                                                0B  12  007D0          BNEQ    86$                                     ;
                             50            54   FC      A0  C1  007D2          ADDL3   -4(R0), R4, R0                          ; 3322
                                           60   3C      A6  D0  007D7          MOVL    60(HEADER), (R0)                        ;
                                                15  11  007DB          BRB     87$                                     ;
                             50            54   FC      A0  C1  007DD 86$:     ADDL3   -4(R0), R4, R0                          ; 3323
                                           53   09      A6  9A  007E2          MOVZBL  9(HEADER), R3                           ;
                             53            53           10  78  007E6          ASHL    #16, R3, R3                             ;
                                           51   08      A6  9A  007EA          MOVZBL  8(HEADER), R1                           ;
                             60            53           51  C1  007EE          ADDL3   R1, R3, (R0)                            ;
                                50 00000000'FF48          DE  007F2 87$:     MOVAL   @ALLOCATION[RVN], R0                    ; 3324
                             50            54   FC      A0  C1  007FA          ADDL3   -4(R0), R4, R0                          ;
                             51            5B 00000000'  EF  C1  007FF          ADDL3   TOTAL_SIZE, EXT_SEQ, R1                 ;
                                           60   01      A1  9E  00807          MOVAB   1(R1), (R0)                             ;
                                50 00000000'FF48          DE  0080B          MOVAL   @USAGE[RVN], R0                         ; 3325
                             50            54   FC      A0  C1  00813          ADDL3   -4(R0), R4, R0                          ;
                             60      08    A2           10  9C  00818          ROTL    #16, 8(FAT), (R0)                       ;
                                           08   A2      D5  00821          TSTL    8(FAT)                                  ; 3327
                                                07  13  00820          BEQL    88$                                     ;
                                           0C   A2      B5  00822          TSTW    12(FAT)                                 ; 3328
                                                02  12  00825          BNEQ    88$                                     ;
                                                60  D7  00827          DECL    (R0)                                    ; 3331
                                      63 00000000'  EF  E9  00829 88$:     BLBC    QUOTA_ACTIVE, 91$                       ; 3337
                                      04            57  D1  00830          CMPL    FILE_NUMBER, #4                         ;
                                                5E  1F  00833          BLSSU   91$                                     ;
                             50            5B 00000000'  EF  C1  00835          ADDL3   TOTAL_SIZE, EXT_SEQ, R0                 ; 3342
                                           01   A0      9F  00839          PUSHAB  1(R0)                                   ;
                                                7E  D4  00840          CLRL    -(SP)                                   ; 3339
                                           3C   A6      DD  00842          PUSHL   60(HEADER)                              ; 3340
                                     0000V  CF      03  FB  00845          CALLS   #3, COUNT_QUOTA                         ;
```

```
                                    47  11 0084A        BRB     91$                                          ; 2889
                 7E                 01  CE 0084C 89$:   MNEGL   #1, -(SP)                                    ; 3350
                            74      AE  9F 0084F        PUSHAB  FILE_ID                                      ;
                                    56  DD 00852        PUSHL   HEADER                                       ;
                          0140      8F  BB 00854        PUSHR   #^M<R6,R8>                                   ;
            0000V   CF              05  FB 00858        CALLS   #5, MAP_PROCESS                              ;
                                    34  11 0085D        BRB     91$                                          ; 2793
                 31                 50  E9 0085F 90$:   BLBC    R0, 91$                                      ; 3359
                 50  00000000'FF    48  DE 00862        MOVAL   @MAP[RVN], R0                                ; 3362
                                    57  D7 0086A        DECL    R7                                           ;
      22     FC  B0                 57  E1 0086C        BBC     R7, @-4(R0), 91$                             ;
                                    56  DD 00871        PUSHL   HEADER                                       ; 3365
                            74      AE  9F 00873        PUSHAB  FILE_ID                                      ;
                     00000000G      8F  DD 00876        PUSHL   #VERIFY$_BADHEADER                           ;
            0000V   CF              03  FB 0087C        CALLS   #3, HEADER_ERROR                             ;
            0000V   CF              00  FB 00881        CALLS   #0, DO_REPAIR                                ; 3366
                 0A                 50  E9 00886        BLBC    R0, 91$                                      ;
                                    56  DD 00889        PUSHL   HEADER                                       ; 3368
                            74      AE  9F 0088B        PUSHAB  FILE_ID                                      ;
            0000V   CF              02  FB 0088E        CALLS   #2, DELETE_HEADER                            ;
                 56       0200      C6  9E 00893 91$:   MOVAB   512(R6), HEADER                              ; 3374
      02     10  AE           14    AE  F2 00898 92$:   AOBLSS  READ_COUNT, XVBN, 93$                        ; 2759
                                    03  11 0089E        BRB     94$                                          ;
                                  F92C  31 008A0 93$:   BRW     16$                                          ;
             08  AE  00000040       8F  C0 008A3 94$:   ADDL2   #64, VBN                                     ; 3378
                                  F7E7  31 008AB        BRW     4$                                           ; 2672
                                    7E  7C 008AE 95$:   CLRQ    -(SP)                                        ; 3386
                                    7E  7C 008B0        CLRQ    -(SP)                                        ;
                                    7E  7C 008B2        CLRQ    -(SP)                                        ;
                                    7E  7C 008B4        CLRQ    -(SP)                                        ;
                 7E                 34  7D 008B6        MOVQ    #52, -(SP)                                   ;
                     00000000'      EF  DD 008B9        PUSHL   CHANNEL                                      ;
                                    7E  D4 008BF        CLRL    -(SP)                                        ;
         00000000G  00              0C  FB 008C1        CALLS   #12, SYS$QIOW                                ;
  F742       58      01       04    AE  F1 008C8 96$:   ACBL    4(SP), #1, RVN, 1$                           ; 2645
                                        04 008CF        RET                                                  ; 3388

; Routine Size:  2256 bytes,    Routine Base:  CODE + 272A
```

```
;  3396     3389  1 ROUTINE VERIFY_HEADER(HEADER,FILE_ID)=
;  3397     3390  1
;  3398     3391  1 !++
;  3399     3392  1 !
;  3400     3393  1 ! FUNCTIONAL DESCRIPTION:
;  3401     3394  1 !       This routine determines if the block given it is a valid file header.
;  3402     3395  1 !
;  3403     3396  1 ! INPUT PARAMETERS:
;  3404     3397  1 !       HEADER          - Pointer to header.
;  3405     3398  1 !       FILE_ID         - Purported file ID.
;  3406     3399  1 !
;  3407     3400  1 ! IMPLICIT INPUTS:
;  3408     3401  1 !       NONE
;  3409     3402  1 !
;  3410     3403  1 ! OUTPUT PARAMETERS:
;  3411     3404  1 !       NONE
;  3412     3405  1 !
;  3413     3406  1 ! IMPLICIT OUTPUTS:
;  3414     3407  1 !       NONE
;  3415     3408  1 !
;  3416     3409  1 ! ROUTINE VALUE:
;  3417     3410  1 !       0 if invalid file header
;  3418     3411  1 !       1 if valid file header
;  3419     3412  1 !       2 if deleted file header
;  3420     3413  1 !
;  3421     3414  1 ! SIDE EFFECTS:
;  3422     3415  1 !       NONE
;  3423     3416  1 !
;  3424     3417  1 !--
;  3425     3418  1
;  3426     3419  2 BEGIN
;  3427     3420  2 MAP
;  3428     3421  2         HEADER:         REF BBLOCK,     ! Pointer to file header
;  3429     3422  2         FILE_ID:        REF BBLOCK;     ! Pointer to file ID
;  3430     3423  2
;  3431     3424  2
;  3432     3425  2 ! First check the structure level.
;  3433     3426  2 !
;  3434     3427  2 IF .HEADER[FH2$B_STRUCLEV] NEQ .STRUCTURE_LEVEL
;  3435     3428  2 THEN
;  3436     3429  2     RETURN 0;
;  3437     3430  2
;  3438     3431  2
;  3439     3432  2 IF .STRUCTURE_LEVEL EQL 2
;  3440     3433  2 THEN
;  3441     3434  3     BEGIN
;  3442     3435  3
;  3443     3436  3     ! Check the area offsets and the retrieval pointer use counts for
;  3444     3437  3     ! consistency.
;  3445     3438  3     !
;  3446     3439  3     IF
;  3447     3440  3         .HEADER[FH2$B_IDOFFSET] LSSU $BYTEOFFSET(FH2$L_HIGHWATER)/2 OR
;  3448     3441  3         .HEADER[FH2$B_MPOFFSET] LSSU .HEADER[FH2$B_IDOFFSET] OR
;  3449     3442  3         .HEADER[FH2$B_ACOFFSET] LSSU .HEADER[FH2$B_MPOFFSET] OR
;  3450     3443  3         .HEADER[FH2$B_RSOFFSET] LSSU .HEADER[FH2$B_ACOFFSET] OR
;  3451     3444  3         .HEADER[FH2$B_MAP_INUSE] GTRU .HEADER[FH2$B_ACOFFSET] - .HEADER[FH2$B_MPOFFSET]
;  3452     3445  3     THEN
```

```
;  3453          3446  1          RETURN 0;
;  3454          3447  1
;  3455          3448  1
;  3456          3449  1      ! At this point, we have verified that the block at least once was a
;  3457          3450  1      ! valid file header.
;  3458          3451  1      !
;  3459          3452  1      ! Look at the file number in the header. If zero, this is a
;  3460          3453  1      ! deleted header.
;  3461          3454  1      !
;  3462          3455  1      IF
;  3463          3456  1          .HEADER[FH2$W_FID_NUM] EQL 0 AND
;  3464          3457  1          .HEADER[FH2$B_FID_NMX] EQL 0
;  3465          3458  1      THEN
;  3466          3459  1          RETURN 2;
;  3467          3460  1
;  3468          3461  1
;  3469          3462  1      ! Now compute the header checksum.
;  3470          3463  1      !
;  3471          3464  1      IF NOT CHECKSUM(.HEADER)
;  3472          3465  1      THEN
;  3473          3466  1          RETURN 2;
;  3474          3467  1
;  3475          3468  1
;  3476          3469  1      ! Check file number and file sequence number.
;  3477          3470  1      !
;  3478          3471  1      IF
;  3479          3472  1          .HEADER[FH2$W_FID_NUM] NEQ .FILE_ID[FID$W_NUM] OR
;  3480          3473  1          .HEADER[FH2$B_FID_NMX] NEQ .FILE_ID[FID$B_NMX] OR
;  3481          3474  1          .HEADER[FH2$W_FID_SEQ] NEQ .FILE_ID[FID$W_SEQ]
;  3482          3475  1      THEN
;  3483          3476  1          RETURN 2;
;  3484          3477  1      END
;  3485          3478  1  ELSE
;  3486          3479  1      BEGIN
;  3487          3480  1      LOCAL
;  3488          3481  1          MAP_AREA:       REF BBLOCK;
;  3489          3482  1
;  3490          3483  1
;  3491          3484  1      ! Check the area offsets, the extension RVN, and the retrieval pointer
;  3492          3485  1      ! data for consistency.
;  3493          3486  1      !
;  3494          3487  1      IF
;  3495          3488  1          .HEADER[FM1$B_IDOFFSET] NEQ FM1$C_LENGTH / 2 OR
;  3496          3489  1          .HEADER[FM1$B_MPOFFSET] NEQ (FM1$C_LENGTH + FI1$C_LENGTH) / 2
;  3497          3490  1      THEN
;  3498          3491  1          RETURN 0;
;  3499          3492  1
;  3500          3493  1
;  3501          3494  1      MAP_AREA = .HEADER + .HEADER[FM1$B_MPOFFSET]*2;
;  3502          3495  1      IF
;  3503          3496  1          .MAP_AREA[FM1$B_EX_RVN] NEQ 0 OR
;  3504          3497  1          .MAP_AREA[FM1$B_COUNTSIZE] NEQ 1 OR
;  3505          3498  1          .MAP_AREA[FM1$B_LBNSIZE] NEQ 3 OR
;  3506          3499  1          .MAP_AREA[FM1$B_INUSE] GTRU .MAP_AREA[FM1$B_AVAIL] OR
;  3507          3500  1          .MAP_AREA[FM1$B_AVAIL] GTRU 255 - (.MAP_AREA + FM1$C_POINTERS - .HEADER) / 2
;  3508          3501  1      THEN
;  3509          3502  1          RETURN 0;
```

```
3
3         ! At this point, we have verified that the block at least once was a
3         ! valid file header.
3         !
3         ! Look at the file number in the header. If zero, this is a
3         ! deleted header.
3         !
3         IF .HEADER[FH1$W_FID_NUM] EQL 0
3         THEN
3             RETURN 2;
3
3
3         ! Now compute the header checksum.
3         !
3         IF NOT CHECKSUM(.HEADER)
3         THEN
3             RETURN 2;
3
3
3         ! Check file number and file sequence number.
3         !
3         IF
3             .HEADER[FH1$W_FID_NUM] NEQ .FILE_ID[FID$W_NUM] OR
3             .HEADER[FH1$W_FID_SEQ] NEQ .FILE_ID[FID$W_SEQ]
3         THEN
3             RETURN 2;
2         END;
2
2
2     ! Header is OK.
2     !
2     RETURN 1;
1     END;
```

```
                                      003C 00000 VERIFY_HEADER:
                                                        .WORD   Save R2,R3,R4,R5                            ; 3389
                  55 00000000'  EF  9E 00002            MOVAB   STRUCTURE_LEVEL, R5                         ;
                  54 00000000G  EF  9E 00009            MOVAB   CHECKSUM, R4                                ;
                  52        04  AC  D0 00010            MOVL    HEADER, R2                                  ; 3427
 65     07  A2    08            00  ED 00014            CMPZV   #0, #8, 7(R2), STRUCTURE_LEVEL              ;
                                03  13 0001A            BEQL    1$                                          ;
                              0080  31 0001C            BRW     7$                                          ;
                  53        08  AC  D0 0001F 1$:        MOVL    FILE_ID, R3                                 ; 3472
                  02            65  D1 00023            CMPL    STRUCTURE_LEVEL, #2                         ; 3432
                                55  12 00026            BNEQ    6$                                          ;
                  26            62  91 00028            CMPB    (R2), #38                                   ; 3440
                                12  1F 0002B            BLSSU   2$                                          ;
                  62        01  A2  91 0002D            CMPB    1(R2), (R2)                                 ; 3441
                                0C  1F 00031            BLSSU   2$                                          ;
              01  A2        02  A2  91 00033            CMPB    2(R2), 1(R2)                                ; 3442
                                05  1F 00038            BLSSU   2$                                          ;
              02  A2        03  A2  91 0003A            CMPB    3(R2), 2(R2)                                ; 3443
```

```
                                        03  1E 0003F 2$:      BGEQU   4$                                    ;
                                      009D  31 00041 3$:      BRW     11$                                   ;
                         51       02    A2  9A 00044 4$:      MOVZBL  2(R2), R1                             ; 3444
                         50       01    A2  9A 00048          MOVZBL  1(R2), R0                             ;
                         51             50  C2 0004C          SUBL2   R0, R1                                ;
      51       3A   A2   08             00  ED 0004F          CMPZV   #0, #8, 58(R2), R1                    ;
                                        EA  1A 00055          BGTRU   3$                                    ;
                                  08    A2  B5 00057          TSTW    8(R2)                                 ; 3456
                                        05  12 0005A          BNEQ    5$                                    ;
                                  0D    A2  95 0005C          TSTB    13(R2)                                ; 3457
                                        78  13 0005F          BEQL    9$                                    ;
                                        52  DD 00061 5$:      PUSHL   R2                                    ; 3464
                         64             01  FB 00063          CALLS   #1, CHECKSUM                          ;
                         70             50  E9 00066          BLBC    R0, 9$                                ;
                         63       08    A2  B1 00069          CMPW    8(R2), (R3)                           ; 3472
                                        6A  12 0006D          BNEQ    9$                                    ;
                    05   A3       0D    A2  91 0006F          CMPB    13(R2), 5(R3)                         ; 3473
                                        63  12 00074          BNEQ    9$                                    ;
                    02   A3       0A    A2  B1 00076          CMPW    10(R2), 2(R3)                         ; 3474
                                        5A  11 0007B          BRB     8$                                    ;
                         17             62  91 0007D 6$:      CMPB    (R2), #23                             ; 3488
                                        5F  12 00080          BNEQ    11$                                   ;
                         2E       01    A2  91 00082          CMPB    1(R2), #46                            ; 3489
                                        59  12 00086          BNEQ    11$                                   ;
                         50       01    A2  9A 00088          MOVZBL  1(R2), R0                             ; 3494
                         50           6240  3E 0008C          MOVAW   (R2)[R0], MAP_AREA                    ;
                                  01    A0  95 00090          TSTB    1(MAP_AREA)                           ; 3496
                                        4C  12 00093          BNEQ    11$                                   ;
                         01       06    A0  91 00095          CMPB    6(MAP_AREA), #1                       ; 3497
                                        46  12 00099          BNEQ    11$                                   ;
                         03       07    A0  91 0009B          CMPB    7(MAP_AREA), #3                       ; 3498
                                        40  12 0009F 7$:      BNEQ    11$                                   ;
                    09   A0       08    A0  91 000A1          CMPB    8(MAP_AREA), 9(MAP_AREA)              ; 3499
                                        39  1A 000A6          BGTRU   11$                                   ;
               51        52             50  C3 000A8          SUBL3   MAP_AREA, R2, R1                      ; 3500
                         51             0A  C2 000AC          SUBL2   #10, R1                               ;
                         51             02  C6 000AF          DIVL2   #2, R1                                ;
                         51     00FF    C1  9E 000B2          MOVAB   255(R1), R1                           ;
      51       09   A0   08             00  ED 000B7          CMPZV   #0, #8, 9(MAP_AREA), R1               ;
                                        22  1A 000BD          BGTRU   11$                                   ;
                                  02    A2  B5 000BF          TSTW    2(R2)                                 ; 3511
                                        15  13 000C2          BEQL    9$                                    ;
                                        52  DD 000C4          PUSHL   R2                                    ; 3518
                         64             01  FB 000C6          CALLS   #1, CHECKSUM                          ;
                         0D             50  E9 000C9          BLBC    R0, 9$                                ;
                         63       02    A2  B1 000CC          CMPW    2(R2), (R3)                           ; 3526
                                        07  12 000D0          BNEQ    9$                                    ;
                    02   A3       04    A2  B1 000D2          CMPW    4(R2), 2(R3)                          ; 3527
                                        04  13 000D7 8$:      BEQL    10$                                   ;
                         50             02  D0 000D9 9$:      MOVL    #2, R0                                ; 3529
                                            04 000DC          RET                                           ;
                         50             01  D0 000DD 10$:     MOVL    #1, R0                                ; 3535
                                            04 000E0          RET                                           ;
                                        50  D4 000E1 11$:     CLRL    R0                                    ; 3536
                                            04 000E3          RET                                           ;

; Routine Size:  228 bytes,    Routine Base:  CODE + 2FFA
```

```
;  3545     3537  1 ROUTINE MAP_PROCESS(HEADER,RVN,MAIN_HEADER,FILE_ID,SCAN_TYPE): NOVALUE=
;  3546     3538  1
;  3547     3539  1 !++
;  3548     3540  1 !
;  3549     3541  1 ! FUNCTIONAL DESCRIPTION:
;  3550     3542  1 !       This routine computes the number of blocks mapped by the specified
;  3551     3543  1 !       file header, and marks them busy in the "new" storage bitmap.
;  3552     3544  1 !
;  3553     3545  1 ! INPUT PARAMETERS:
;  3554     3546  1 !       HEADER          - Pointer to file header to be processed.
;  3555     3547  1 !       RVN             - Relative volume number of header.
;  3556     3548  1 !       MAIN_HEADER     - Pointer to file header for segment 0.
;  3557     3549  1 !       FILE_ID         - Pointer to file ID of main header.
;  3558     3550  1 !       SCAN_TYPE       - +1:  Primary header in file number order
;  3559     3551  1 !                           0:  Extension header in extension linkage order
;  3560     3552  1 !                          -1:  Extension header in file number order
;  3561     3553  1 !
;  3562     3554  1 ! IMPLICIT INPUTS:
;  3563     3555  1 !       TOTAL_SIZE      - Blocks mapped thus far.
;  3564     3556  1 !
;  3565     3557  1 ! OUTPUT PARAMETERS:
;  3566     3558  1 !       NONE
;  3567     3559  1 !
;  3568     3560  1 ! IMPLICIT OUTPUTS:
;  3569     3561  1 !       Clusters marked busy in "new" storage bitmap.
;  3570     3562  1 !       TOTAL_SIZE      - Updated.
;  3571     3563  1 !
;  3572     3564  1 ! ROUTINE VALUE:
;  3573     3565  1 !       Number of blocks in header.
;  3574     3566  1 !
;  3575     3567  1 ! SIDE EFFECTS:
;  3576     3568  1 !       NONE
;  3577     3569  1 !
;  3578     3570  1 !--
;  3579     3571  1
;  3580     3572  2 BEGIN
;  3581     3573  2 MAP
;  3582     3574  2         HEADER:         REF BBLOCK;     ! File header
;  3583     3575  2 LINKAGE
;  3584     3576  2         L_MAP_POINTER=  JSB:
;  3585     3577  2                         GLOBAL(COUNT=6, LBN=7, MAP_POINTER=8);
;  3586     3578  2 GLOBAL REGISTER
;  3587     3579  2         COUNT=          6,              ! Retrieval pointer count
;  3588     3580  2         LBN=            7,              ! Retrieval pointer LBN
;  3589     3581  2         MAP_POINTER=    8: REF BBLOCK;  ! Pointer to scan map area
;  3590     3582  2 EXTERNAL ROUTINE
;  3591     3583  2         GET_MAP_POINTER: L_MAP_POINTER; ! Get value of ODS-2 file map pointer
;  3592     3584  2 LOCAL
;  3593     3585  2         END_MAP_USED,                   ! Pointer to end of used map area
;  3594     3586  2         END_MAP_ALLOC;                  ! Pointer to end of allocated map area
;  3595     3587  2
;  3596     3588  2
;  3597     3589  2 ! Get pointers to the map area, the end of the used portion of the map area,
;  3598     3590  2 ! and the end of the allocated portion of the map area.
;  3599     3591  2 !
;  3600     3592  2 IF .STRUCTURE_LEVEL EQL 2
;  3601     3593  2 THEN
```

```
;  3602     3594  3      BEGIN
;  3603     3595  3      MAP_POINTER = .HEADER + .HEADER[FH2$B_MPOFFSET]*2;
;  3604     3596  3      END_MAP_ALLOC = .HEADER + .HEADER[FH2$B_ACOFFSET]*2;
;  3605     3597  3      END_MAP_USED = .MAP_POINTER + .HEADER[FH2$B_MAP_INUSE]*2;
;  3606     3598  3      END
;  3607     3599  2  ELSE
;  3608     3600  2      BEGIN
;  3609     3601  2      MAP_POINTER = .HEADER + .HEADER[FH1$B_MPOFFSET]*2;
;  3610     3602  2      END_MAP_ALLOC = .HEADER + $BYTEOFFSET(FH1$W_CHECKSUM);
;  3611     3603  2      END_MAP_USED = .MAP_POINTER + FM1$C_POINTERS + .MAP_POINTER[FM1$B_INUSE]*2;
;  3612     3604  2      MAP_POINTER = .MAP_POINTER + FM1$C_POINTERS;
;  3613     3605  2      END;
;  3614     3606  2
;  3615     3607  2
;  3616     3608  2  ! Loop until entire map processed.
;  3617     3609  2  !
;  3618     3610  2  UNTIL .MAP_POINTER GEQA .END_MAP_USED DO
;  3619     3611  2      BEGIN
;  3620     3612  2      LOCAL
;  3621     3613  2          DIVIDEND:       VECTOR[2],      ! Quadword for EDIV dividend
;  3622     3614  2          REMAINDER,                      ! EDIV remainder
;  3623     3615  2          CLUSTER_NUMBER,                 ! Cluster bit number
;  3624     3616  2          CLUSTER_COUNT;                  ! Count of clusters to mark allocated
;  3625     3617  2
;  3626     3618  2
;  3627     3619  2      ! Get count and LBN.
;  3628     3620  2      !
;  3629     3621  2      IF .STRUCTURE_LEVEL EQL 2
;  3630     3622  2      THEN
;  3631     3623  2          GET_MAP_POINTER()
;  3632     3624  2      ELSE
;  3633     3625  4          BEGIN
;  3634     3626  4          LBN = .MAP_POINTER[FM1$W_LOWLBN];
;  3635     3627  4          LBN<16,8> = .MAP_POINTER[FM1$B_HIGHLBN];
;  3636     3628  4          COUNT = .MAP_POINTER[FM1$B_COUNT] + 1;
;  3637     3629  4          MAP_POINTER = .MAP_POINTER + 4;
;  3638     3630  3          END;
;  3639     3631  3
;  3640     3632  3
;  3641     3633  3      ! Convert count and LBN by cluster factor.  Make sure that the values are
;  3642     3634  3      ! even multiples of the cluster factor.
;  3643     3635  3      !
;  3644     3636  3      DIVIDEND[1] = 0;
;  3645     3637  3      DIVIDEND[0] = .COUNT;
;  3646     3638  3      EDIV(CLUSTER_FACTOR[.RVN-1], DIVIDEND, CLUSTER_COUNT, REMAINDER);
;  3647     3639  3      IF .REMAINDER NEQ 0
;  3648     3640  3      THEN
;  3649     3641  3          HEADER_ERROR(
;  3650     3642  3              VERIFY$_MAPAREA,
;  3651     3643  3              .FILE_ID,
;  3652     3644  3              .MAIN_HEADER);
;  3653     3645  3      DIVIDEND[0] = .LBN;
;  3654     3646  3      EDIV(CLUSTER_FACTOR[.RVN-1], DIVIDEND, CLUSTER_NUMBER, REMAINDER);
;  3655     3647  3      IF .REMAINDER NEQ 0
;  3656     3648  3      THEN
;  3657     3649  3          HEADER_ERROR(
;  3658     3650  3              VERIFY$_MAPAREA,
```

```
;  3659     3651  3                .FILE_ID,
;  3660     3652  3                .MAIN_HEADER);
;  3661     3653  3
;  3662     3654  3
;  3663     3655  3        ! Loop for each bit of the bitmap that is affected.  If we are doing the
;  3664     3656  3        ! multiple allocation scan, report any bits that are multiply allocated.
;  3665     3657  3        ! Otherwise, mark the bits allocated, and if they are already marked,
;  3666     3658  3        ! mark them in the multiple allocation map and set the multiple
;  3667     3659  3        ! allocation summary flag.
;  3668     3660  3        !
;  3669     3661  3        IF .DUAL_ALLOC_PASS
;  3670     3662  3        THEN
;  3671     3663  3            INCR J FROM 0 TO .CLUSTER_COUNT-1 DO
;  3672     3664  4                BEGIN
;  3673     3665  4                IF .CLUSTER_NUMBER + .J GTRU .SMAP_SIZE[.RVN-1]*512*8-1
;  3674     3666  4                THEN
;  3675     3667  4                    EXITLOOP;
;  3676     3668  4
;  3677     3669  4
;  3678     3670  4                IF
;  3679     3671  4                    .BITVECTOR[.MULTSMAP[.RVN-1], .CLUSTER_NUMBER + .J] AND
;  3680     3672  4                    .SCAN_TYPE
;  3681     3673  4                THEN
;  3682     3674  5                    BEGIN
;  3683     3675  5                    HEADER_ERROR(
;  3684     3676  5                        VERIFY$_MULTALLOC,
;  3685     3677  5                        .FILE_ID,
;  3686     3678  5                        .MAIN_HEADER,
;  3687     3679  5                        1 + .TOTAL_SIZE + .J * .CLUSTER_FACTOR[.RVN-1],
;  3688     3680  5                        .TOTAL_SIZE + (.J + 1) * .CLUSTER_FACTOR[.RVN-1],
;  3689     3681  5                        .LBN + .J * .CLUSTER_FACTOR[.RVN-1],
;  3690     3682  5                        .LBN + (.J + 1) * .CLUSTER_FACTOR[.RVN-1] - 1,
;  3691     3683  5                        .RVN);
;  3692     3684  4                    END;
;  3693     3685  4                END
;  3694     3686  3        ELSE
;  3695     3687  3            INCR J FROM .CLUSTER_NUMBER TO .CLUSTER_NUMBER + .CLUSTER_COUNT - 1 DO
;  3696     3688  4                BEGIN
;  3697     3689  4                IF .J GTRU .SMAP_SIZE[.RVN-1]*512*8-1
;  3698     3690  4                THEN
;  3699     3691  5                    BEGIN
;  3700     3692  5                    HEADER_ERROR(
;  3701     3693  5                        VERIFY$_MAPAREA,
;  3702     3694  5                        .FILE_ID,
;  3703     3695  5                        .MAIN_HEADER);
;  3704     3696  5                    EXITLOOP;
;  3705     3697  4                    END;
;  3706     3698  4
;  3707     3699  4
;  3708     3700  4                IF .SCAN_TYPE GEQ 0
;  3709     3701  4                THEN
;  3710     3702  4                    BITVECTOR[.NSMAP[.RVN-1], .J] = FALSE;
;  3711     3703  4
;  3712     3704  4
;  3713     3705  4                IF .SCAN_TYPE
;  3714     3706  4                THEN
;  3715     3707  4                    IF TESTBITCC(BITVECTOR[.VSMAP[.RVN-1], .J])
```

```
;  3716     3708  4                    THEN
;  3717     3709  5                        BEGIN
;  3718     3710  5                        BITVECTOR[.MULTSMAP[.RVN-1], .J] = TRUE;
;  3719     3711  5                        DUAL_ALLOC_FOUND = TRUE;
;  3720     3712  4                        END;
;  3721     3713  3                END;
;  3722     3714  3
;  3723     3715  3
;  3724     3716  3            ! Finally, add the count into the total file size.
;  3725     3717  3            !
;  3726     3718  3            TOTAL_SIZE = .TOTAL_SIZE + .COUNT;
;  3727     3719  2            END;
;  3728     3720  2
;  3729     3721  2
;  3730     3722  2
;  3731     3723  2        ! Make sure that the last map pointer ended at the location identified by
;  3732     3724  2        ! MAP_INUSE.
;  3733     3725  2        !
;  3734     3726  2        IF .MAP_POINTER NEQA .END_MAP_USED
;  3735     3727  2        THEN
;  3736     3728  2            HEADER_ERROR(
;  3737     3729  2                VERIFY$_MAPAREA,
;  3738     3730  2                .FILE_ID,
;  3739     3731  2                .MAIN_HEADER);
;  3740     3732  2
;  3741     3733  2
;  3742     3734  2        ! Make sure that the unused portion of the map area is zero.
;  3743     3735  2        !
;  3744     3736  2        IF .END_MAP_ALLOC GTRA .END_MAP_USED
;  3745     3737  2        THEN
;  3746     3738  2            IF CH$FIND_NOT_CH(.END_MAP_ALLOC-.END_MAP_USED, .END_MAP_USED, 0) NEQ 0
;  3747     3739  2            THEN
;  3748     3740  2                HEADER_ERROR(
;  3749     3741  2                    VERIFY$_MAPAREA,
;  3750     3742  2                    .FILE_ID,
;  3751     3743  2                    .MAIN_HEADER);
;  3752     3744  1 END;


                                                .EXTRN  GET_MAP_POINTER

                                OFFC 00000 MAP_PROCESS:
                                                .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11            ; 3537
5B 00000000'   EF   9E 00002                    MOVAB   CLUSTER_FACTOR, R11
5E             08   C2 00009                    SUBL2   #8, SP
50       04    AC   D0 0000C                    MOVL    HEADER, R0                                      ; 3595
51       01    A0   9E 00010                    MOVAB   1(R0), R1                                       ; 3592
02     FF70    CB   D1 00014                    CMPL    STRUCTURE_LEVEL, #2
               19   12 00019                    BNEQ    1$
51             61   9A 0001B                    MOVZBL  (R1), R1                                        ; 3595
58          6041   3E 0001E                    MOVAW   (R0)[R1], MAP_POINTER
51       02    A0   9A 00022                    MOVZBL  2(R0), R1                                       ; 3596
54          6041   3E 00026                    MOVAW   (R0)[R1], END_MAP_ALLOC
50       3A    A0   9A 0002A                    MOVZBL  58(R0), R0                                      ; 3597
59          6840   3E 0002E                    MOVAW   (MAP_POINTER)[R0], END_MAP_USED
               18   11 00032                    BRB     2$                                              ; 3592
```

```
                       51            61  9A 00034 1$:    MOVZBL  (R1), R1                                          ; 3601
                       58          6041  3E 00037        MOVAW   (R0)[R1], MAP_POINTER                             ;
                       54     01FE   C0  9E 0003B        MOVAB   510(R0), END_MAP_ALLOC                            ; 3602
                       50       08   A8  9A 00040        MOVZBL  8(MAP_POINTER), R0                                ; 3603
                       59       0A A840  3E 00044        MOVAW   10(MAP_POINTER)[R0], END_MAP_USED                 ;
                       58            0A  C0 00049        ADDL2   #10, MAP_POINTER                                  ; 3604
         55      08    AC            01  C3 0004C 2$:    SUBL3   #1, RVN, R5                                       ; 3638
                       55            04  C4 00051        MULL2   #4, R5                                            ;
                       59            58  D1 00054 3$:    CMPL    MAP_POINTER, END_MAP_USED                         ; 3610
                                     03  1F 00057        BLSSU   4$                                                ;
                                   013A  31 00059        BRW     19$                                               ;
                       02     FF70   CB  D1 0005C 4$:    CMPL    STRUCTURE_LEVEL, #2                               ; 3621
                                     08  12 00061        BNEQ    5$                                                ;
                          00000000G  EF  16 00063        JSB     GET_MAP_POINTER                                   ; 3623
                                     0F  11 00069        BRB     6$                                                ;
                       57       02   A8  3C 0006B 5$:    MOVZWL  2(MAP_POINTER), LBN                               ; 3626
    57           08    10            88  F0 0006F        INSV    (MAP_POINTER)+, #16, #8, LBN                      ; 3627
                       56       FD   A8  9A 00074        MOVZBL  -3(MAP_POINTER), COUNT                            ; 3628
                                     56  D6 00078        INCL    COUNT                                             ;
                                04   AE  D4 0007A 6$:    CLRL    DIVIDEND+4                                        ; 3636
                       6E            56  D0 0007D        MOVL    COUNT, DIVIDEND                                   ; 3637
                 50    6B            55  C1 00080        ADDL3   R5, CLUSTER_FACTOR, R0                            ; 3638
    53           5A    6E            60  7B 00084        EDIV    (R0), DIVIDEND, CLUSTER_COUNT, REMAINDER          ;
                                     53  D5 00089        TSTL    REMAINDER                                         ; 3639
                                     11  13 0008B        BEQL    7$                                                ;
                                0C   AC  DD 0008D        PUSHL   MAIN_HEADER                                       ; 3644
                                10   AC  DD 00090        PUSHL   FILE_ID                                           ; 3643
                          00000000G  8F  DD 00093        PUSHL   #VERIFY$_MAPAREA                                  ; 3641
                     0000V     CF    03  FB 00099        CALLS   #3, HEADER_ERROR                                  ;
                       6E            57  D0 0009E 7$:    MOVL    LBN, DIVIDEND                                     ; 3645
                 50    6B            55  C1 000A1        ADDL3   R5, CLUSTER_FACTOR, R0                            ; 3646
    53           52    6E            60  7B 000A5        EDIV    (R0), DIVIDEND, CLUSTER_NUMBER, REMAINDER         ;
                                     53  D5 000AA        TSTL    REMAINDER                                         ; 3647
                                     11  13 000AC        BEQL    8$                                                ;
                                0C   AC  DD 000AE        PUSHL   MAIN_HEADER                                       ; 3652
                                10   AC  DD 000B1        PUSHL   FILE_ID                                           ; 3651
                          00000000G  8F  DD 000B4        PUSHL   #VERIFY$_MAPAREA                                  ; 3649
                     0000V     CF    03  FB 000BA        CALLS   #3, HEADER_ERROR                                  ;
                       69 00000000'  EF  E9 000BF 8$:    BLBC    DUAL_ALLOC_PASS, 11$                              ; 3661
                       53            01  CE 000C6        MNEGL   #1, J                                             ; 3663
                                     5E  11 000C9        BRB     10$                                               ;
                 51    52            53  C1 000CB 9$:    ADDL3   J, CLUSTER_NUMBER, R1                             ; 3665
                 50    55       F0   AB  C1 000CF        ADDL3   SMAP_SIZE, R5, R0                                 ;
                 50    60            0C  78 000D4        ASHL    #12, (R0), R0                                     ;
                                     50  D7 000D8        DECL    R0                                                ;
                 50                  51  D1 000DA        CMPL    R1, R0                                            ;
                                     79  1A 000DD        BGTRU   13$                                               ;
                 50    55       FC   AB  C1 000DF        ADDL3   MULTSMAP, R5, R0                                  ; 3671
                 40    00  B0        51  E1 000E4        BBC     R1, @0(R0), 10$                                   ;
                       3C       14   AC  E9 000E9        BLBC    SCAN_TYPE, 10$                                    ; 3672
                                08   AC  DD 000ED        PUSHL   RVN                                               ; 3683
                 50             01   A3  9E 000F0        MOVAB   1(R3), R0                                         ; 3682
                 51    6B            55  C1 000F4        ADDL3   R5, CLUSTER_FACTOR, R1                            ;
                 50                  61  C4 000F8        MULL2   (R1), R0                                          ;
                                FF A047  9F 000FB        PUSHAB  -1(R0)[LBN]                                       ;
                 51    53            61  C5 000FF        MULL3   (R1), J, R1                                       ; 3681
                                   6147  9F 00103        PUSHAB  (R1)[LBN]                                         ;
```

```
                00000000'FF40   9F 00106          PUSHAB  @TOTAL_SIZE[R0]                         ; 3680
          50          51 00000000'   EF C1 00108          ADDL3   TOTAL_SIZE, R1, R0                      ; 3679
                       01   A0 9F 0010D          PUSHAB  1(R0)                                   ;
                       0C   AC DD 00110          PUSHL   MAIN_HEADER                             ; 3678
                       10   AC DD 00113          PUSHL   FILE_ID                                 ; 3677
                00000000G   8F DD 00116          PUSHL   #VERIFY$_MULTALLOC                      ; 3675
              0000V CF              08 FB 0011C          CALLS   #8, HEADER_ERROR                        ;
          9E          53              5A F2 00121 10$:     AOBLSS  CLUSTER_COUNT, J, 9$                    ; 3663
                                      5D 11 00125          BRB     18$                             ;
          53          52              5A C1 00127 11$:     ADDL3   CLUSTER_COUNT, CLUSTER_NUMBER, R3       ; 3687
                                      52 D7 0012B          DECL    J                               ; 3700
                                      51 11 0012D          BRB     17$                             ;
          50          55      F0      AB C1 0012F 12$:     ADDL3   SMAP_SIZE, R5, R0               ; 3689
          50          60              0C 78 00134          ASHL    #12, (R0), R0                   ;
                                      50 D7 00138          DECL    R0                              ;
                      50              52 D1 0013A          CMPL    J, R0                           ;
                                      13 1B 0013D          BLEQU   14$                             ;
                       0C   AC DD 0013F          PUSHL   MAIN_HEADER                             ; 3695
                       10   AC DD 00142          PUSHL   FILE_ID                                 ; 3694
                00000000G   8F DD 00145          PUSHL   #VERIFY$_MAPAREA                        ; 3692
              0000V CF              03 FB 0014B          CALLS   #3, HEADER_ERROR                        ;
                                      32 11 00150 13$:     BRB     18$                             ; 3691
                       14   AC D5 00152 14$:     TSTL    SCAN_TYPE                               ; 3700
                                      0A 19 00155          BLSS    15$                             ;
          50          55      F8      AB C1 00157          ADDL3   NSMAP, R5, R0                   ; 3702
          00          00      B0      52 E5 0015C          BBCC    J, @0(R0), 15$                  ;
                              1B      14 AC E9 00161 15$:  BLBC    SCAN_TYPE, 17$                  ; 3705
          50          55      F4      AB C1 00165          ADDL3   VSMAP, R5, R0                   ; 3707
          11          00      B0      52 E4 0016A          BBSC    J, @0(R0), 17$                  ;
          50          55      FC      AB C1 0016F          ADDL3   MULTSMAP, R5, R0                ; 3710
          00          00      B0      52 E2 00174          BBSS    J, @0(R0), 16$                  ;
                00000000'     EF              01 D0 00179 16$:     MOVL    #1, DUAL_ALLOC_FOUND            ; 3711
              AB          52              53 F2 00180 17$:     AOBLSS  R3, J, 12$                      ; 3687
                00000000'     EF              56 C0 00184 18$:     ADDL2   COUNT, TOTAL_SIZE               ; 3718
                                          FEBE 31 00189          BRW     3$                              ; 3610
                                      11  13 0018C 19$:     BEQL    20$                             ; 3726
                                   0C   AC DD 0018E          PUSHL   MAIN_HEADER                     ; 3731
                                   10   AC DD 00191          PUSHL   FILE_ID                         ; 3730
                            00000000G   8F DD 00194          PUSHL   #VERIFY$_MAPAREA                ; 3728
                          0000V CF              03 FB 0019A          CALLS   #3, HEADER_ERROR                ;
                                      59  54 D1 0019F 20$:     CMPL    END_MAP_ALLOC, END_MAP_USED     ; 3736
                                      20  1B 001A2          BLEQU   22$                             ;
                                      54  59 C2 001A4          SUBL2   END_MAP_USED, R4                ; 3738
                              69      54  00 3B 001A7          SKPC    #0, R4, (END_MAP_USED)          ;
                                      02  12 001AB          BNEQ    21$                             ;
                                      51  D4 001AD          CLRL    R1                              ;
                                      51  D5 001AF 21$:     TSTL    R1                              ;
                                      11  13 001B1          BEQL    22$                             ;
                                   0C   AC DD 001B3          PUSHL   MAIN_HEADER                     ; 3743
                                   10   AC DD 001B6          PUSHL   FILE_ID                         ; 3742
                            00000000G   8F DD 001B9          PUSHL   #VERIFY$_MAPAREA                ; 3740
                          0000V CF              03 FB 001BF          CALLS   #3, HEADER_ERROR                ;
                                          04  001CE 22$:     RET                                     ; 3744

; Routine Size:  463 bytes,    Routine Base:  CODE + 30DE
```

```
;  3754     3745  1 ROUTINE CREATE_WINDOW(P_HEADER,P_RVN)=
;  3755     3746  1
;  3756     3747  1 !++
;  3757     3748  1 !
;  3758     3749  1 ! FUNCTIONAL DESCRIPTION:
;  3759     3750  1 !       This routine generates a window block (or blocks) from a file
;  3760     3751  1 !       header, reading the extension headers as necessary.
;  3761     3752  1 !
;  3762     3753  1 ! INPUT PARAMETERS:
;  3763     3754  1 !       P_HEADER        - Pointer to file header to be processed.
;  3764     3755  1 !       P_RVN           - Relative volume number of file header.
;  3765     3756  1 !
;  3766     3757  1 ! IMPLICIT INPUTS:
;  3767     3758  1 !       NONE
;  3768     3759  1 !
;  3769     3760  1 ! OUTPUT PARAMETERS:
;  3770     3761  1 !       NONE
;  3771     3762  1 !
;  3772     3763  1 ! IMPLICIT OUTPUTS:
;  3773     3764  1 !       NONE
;  3774     3765  1 !
;  3775     3766  1 ! ROUTINE VALUE:
;  3776     3767  1 !       Pointer to window block.  If the header maps no space, 0.
;  3777     3768  1 !
;  3778     3769  1 ! SIDE EFFECTS:
;  3779     3770  1 !       NONE
;  3780     3771  1 !
;  3781     3772  1 !--
;  3782     3773  1
;  3783     3774  2 BEGIN
;  3784     3775  2 LINKAGE
;  3785     3776  2         L_MAP_POINTER=  JSB:
;  3786     3777  2                         GLOBAL(COUNT=6, LBN=7, MAP_POINTER=8);
;  3787     3778  2 EXTERNAL ROUTINE
;  3788     3779  2         GET_MAP_POINTER: L_MAP_POINTER; ! Get value of ODS-2 file map pointer
;  3789     3780  2 LOCAL
;  3790     3781  2         HEADER:         REF BBLOCK,     ! Pointer to current file header
;  3791     3782  2         RVN,                            ! RVN of current file header
;  3792     3783  2         EXT_FILE_ID:    BBLOCK[FID$C_LENGTH],   ! Extension file ID
;  3793     3784  2         LOCAL_HEADER:   BBLOCK[512],    ! Local area for file header
;  3794     3785  2         WINDOW_LIST:    VECTOR[2],      ! List head of window list
;  3795     3786  2         WINDOW:         BBLOCK[WDW_S_HEADER + WDW_K_MAXENTRY * WDW_S_ENTRY],
;  3796     3787  2         P:              REF BBLOCK;     ! Pointer to current window entry
;  3797     3788  2
;  3798     3789  2
;  3799     3790  2 ! Initialize.
;  3800     3791  2 !
;  3801     3792  2 HEADER = .P_HEADER;
;  3802     3793  2 RVN = .P_RVN;
;  3803     3794  2 WINDOW_LIST[0] = WINDOW_LIST[1] = 0;
;  3804     3795  2 WINDOW[WDW_LINK] = 0;
;  3805     3796  2 WINDOW[WDW_SIZE] = 0;
;  3806     3797  2 P = WINDOW + WDW_S_HEADER - WDW_S_ENTRY;
;  3807     3798  2
;  3808     3799  2
;  3809     3800  2 ! Loop over this header and all of its extension headers.
;  3810     3801  2 !
```

```
;  3811     3802  2 WHILE TRUE DO
;  3812     3803  2     BEGIN
;  3813     3804  2     GLOBAL REGISTER
;  3814     3805  2         COUNT=          6,              ! Retrieval pointer count
;  3815     3806  2         LBN=            7,              ! Retrieval pointer LBN
;  3816     3807  2         MAP_POINTER=    8: REF BBLOCK;  ! Pointer to scan map area
;  3817     3808  2     LOCAL
;  3818     3809  2         END_MAP;                        ! Pointer to end of used map area
;  3819     3810  2
;  3820     3811  2
;  3821     3812  2     ! Get pointers to the map area and the end of the used portion
;  3822     3813  2     ! of the map area.
;  3823     3814  2     !
;  3824     3815  2     IF .STRUCTURE_LEVEL EQL 2
;  3825     3816  2     THEN
;  3826     3817  3         BEGIN
;  3827     3818  3         MAP_POINTER = .HEADER + .HEADER[FH2$B_MPOFFSET]*2;
;  3828     3819  3         END_MAP = .MAP_POINTER + .HEADER[FH2$B_MAP_INUSE]*2;
;  3829     3820  3         END
;  3830     3821  2     ELSE
;  3831     3822  3         BEGIN
;  3832     3823  3         MAP_POINTER = .HEADER + .HEADER[FH1$B_MPOFFSET]*2;
;  3833     3824  3         END_MAP = .MAP_POINTER + FM1$C_POINTERS + .MAP_POINTER[FM1$B_INUSE]*2;
;  3834     3825  3         MAP_POINTER = .MAP_POINTER + FM1$C_POINTERS;
;  3835     3826  3         END;
;  3836     3827  2
;  3837     3828  2
;  3838     3829  2     ! Loop until entire map processed.
;  3839     3830  2     !
;  3840     3831  2     UNTIL .MAP_POINTER GEQA .END_MAP DO
;  3841     3832  3         BEGIN
;  3842     3833  3
;  3843     3834  3         ! Get count and LBN.
;  3844     3835  3         !
;  3845     3836  3         IF .STRUCTURE_LEVEL EQL 2
;  3846     3837  3         THEN
;  3847     3838  3             GET_MAP_POINTER()
;  3848     3839  3         ELSE
;  3849     3840  4             BEGIN
;  3850     3841  4             LBN = .MAP_POINTER[FM1$W_LOWLBN];
;  3851     3842  4             LBN<16,8> = .MAP_POINTER[FM1$B_HIGHLBN];
;  3852     3843  4             COUNT = .MAP_POINTER[FM1$B_COUNT] + 1;
;  3853     3844  4             MAP_POINTER = .MAP_POINTER + 4;
;  3854     3845  3             END;
;  3855     3846  3
;  3856     3847  3
;  3857     3848  3         ! Collapse with previous map pointer if contiguous with it --
;  3858     3849  3         ! otherwise, generate new map pointer.
;  3859     3850  3         !
;  3860     3851  3         IF
;  3861     3852  4             BEGIN
;  3862     3853  4             IF .WINDOW[WDW_SIZE] NEQ 0
;  3863     3854  4             THEN
;  3864     3855  4                 .P[WDW_COUNT] + .P[WDW_LBN] EQL .LBN
;  3865     3856  4             ELSE
;  3866     3857  4                 FALSE
;  3867     3858  4             END
```

```
;  3868     3859  4          THEN
;  3869     3860  4              P[WDW_COUNT] = .P[WDW_COUNT] + .COUNT
;  3870     3861  4          ELSE
;  3871     3862  5              BEGIN
;  3872     3863  5              IF .WINDOW[WDW_SIZE] GEQU WDW_K_MAXENTRY
;  3873     3864  5              THEN
;  3874     3865  6                  BEGIN
;  3875     3866  6                  LOCAL
;  3876     3867  6                      STATUS,             ! Status return
;  3877     3868  6                      DYNWDW;             ! Dynamic window pointer
;  3878     3869  6
;  3879     3870  6
;  3880     3871  6                  ! Window block has overflowed.  Move local window block to
;  3881     3872  6                  ! dynamic space and initialize for new block.
;  3882     3873  6                  !
;  3883     3874  6                  STATUS = LIB$GET_VM(
;  3884     3875  6                      UPLIT(WDW_S_HEADER + WDW_K_MAXENTRY * WDW_S_ENTRY),
;  3885     3876  6                      DYNWDW);
;  3886     3877  6                  IF NOT .STATUS THEN SIGNAL(VERIFY$_ALLOCMEM, 0, .STATUS);
;  3887     3878  6                  CH$MOVE(
;  3888     3879  6                      WDW_S_HEADER + WDW_K_MAXENTRY * WDW_S_ENTRY,
;  3889     3880  6                      WINDOW,
;  3890     3881  6                      .DYNWDW);
;  3891     3882  6                  IF .WINDOW_LIST[1] NEQ 0
;  3892     3883  6                      THEN BBLOCK[.WINDOW_LIST[1], WDW_LINK] = .DYNWDW;
;  3893     3884  6                  IF .WINDOW_LIST[0] EQL 0 THEN WINDOW_LIST[0] = .DYNWDW;
;  3894     3885  6                  WINDOW_LIST[1] = .DYNWDW;
;  3895     3886  6                  WINDOW[WDW_LINK] = 0;
;  3896     3887  6                  WINDOW[WDW_SIZE] = 0;
;  3897     3888  6                  P = WINDOW + WDW_S_HEADER - WDW_S_ENTRY;
;  3898     3889  5                  END;
;  3899     3890  5
;  3900     3891  5
;  3901     3892  5                  ! Generate new pointer.
;  3902     3893  5                  !
;  3903     3894  5                  WINDOW[WDW_SIZE] = .WINDOW[WDW_SIZE] + 1;
;  3904     3895  5                  P = .P + WDW_S_ENTRY;
;  3905     3896  5                  P[WDW_COUNT] = .COUNT;
;  3906     3897  5                  P[WDW_LBN] = .LBN;
;  3907     3898  5                  END;
;  3908     3899  4              END;
;  3909     3900  4
;  3910     3901  4
;  3911     3902  4          ! If no extension header exists, finish up.
;  3912     3903  4          !
;  3913     3904  4          IF
;  3914     3905  4              BEGIN
;  3915     3906  4              IF .STRUCTURE_LEVEL EQL 2
;  3916     3907  4              THEN
;  3917     3908  4                  .HEADER[FH2$W_EX_FIDNUM] EQL 0
;  3918     3909  4              ELSE
;  3919     3910  5                  BEGIN
;  3920     3911  5                  MAP_POINTER = .HEADER + .HEADER[FH2$B_MPOFFSET]*2;
;  3921     3912  5                  .MAP_POINTER[FM1$W_EX_FILNUM] EQL 0
;  3922     3913  5                  END
;  3923     3914  4              END
;  3924     3915  3          THEN
```

```
;  3925     3916  3          EXITLOOP;
;  3926     3917  3
;  3927     3918  3
;  3928     3919  3      ! Get clean file number and RVN.
;  3929     3920  3      !
;  3930     3921  3      IF .STRUCTURE_LEVEL EQL 2
;  3931     3922  3      THEN
;  3932     3923  4          BEGIN
;  3933     3924  4          EXT_FILE_ID[FID$W_NUM] = .HEADER[FH2$W_EX_FIDNUM];
;  3934     3925  4          EXT_FILE_ID[FID$W_SEQ] = .HEADER[FH2$W_EX_FIDSEQ];
;  3935     3926  4          EXT_FILE_ID[FID$W_RVN] = .HEADER[FH2$W_EX_FIDRVN];
;  3936     3927  4          END
;  3937     3928  3      ELSE
;  3938     3929  4          BEGIN
;  3939     3930  4          EXT_FILE_ID[FID$W_NUM] = .MAP_POINTER[FM1$W_EX_FILNUM];
;  3940     3931  4          EXT_FILE_ID[FID$W_SEQ] = .MAP_POINTER[FM1$W_EX_FILSEQ];
;  3941     3932  4          EXT_FILE_ID[FID$W_RVN] = 1;
;  3942     3933  4          END;
;  3943     3934  3      IF .EXT_FILE_ID[FID$B_RVN] EQL 0 THEN EXT_FILE_ID[FID$B_RVN] = .RVN;
;  3944     3935  3
;  3945     3936  3
;  3946     3937  3      ! Set up header and RVN for next trip through loop.
;  3947     3938  3      !
;  3948     3939  3      HEADER = LOCAL_HEADER;
;  3949     3940  3      RVN = .EXT_FILE_ID[FID$B_RVN];
;  3950     3941  3
;  3951     3942  3
;  3952     3943  3      ! Read extension file header.  If this fails,
;  3953     3944  3      ! exit the loop.
;  3954     3945  3      !
;  3955     3946  3      IF NOT READ_HEADER(EXT_FILE_ID, .HEADER)
;  3956     3947  3      THEN
;  3957     3948  3          EXITLOOP;
;  3958     3949  3      END;
;  3959     3950  3
;  3960     3951  3
;  3961     3952  3  ! Get dynamic window block for current window.
;  3962     3953  3  !
;  3963     3954  3  IF .WINDOW[WDW_SIZE] NEQ 0
;  3964     3955  3  THEN
;  3965     3956  3      BEGIN
;  3966     3957  3      LOCAL
;  3967     3958  3          STATUS,          ! Status return
;  3968     3959  3          DYNWDW;          ! Dynamic window pointer
;  3969     3960  3
;  3970     3961  3
;  3971     3962  3      STATUS = LIB$GET_VM(
;  3972     3963  3          %REF(WDW_S_HEADER + .WINDOW[WDW_SIZE] * WDW_S_ENTRY),
;  3973     3964  3          DYNWDW);
;  3974     3965  3      IF NOT .STATUS THEN SIGNAL(VERIFY$_ALLOCMEM, 0, .STATUS);
;  3975     3966  3      CH$MOVE(
;  3976     3967  3          WDW_S_HEADER + .WINDOW[WDW_SIZE] * WDW_S_ENTRY,
;  3977     3968  3          WINDOW,
;  3978     3969  3          .DYNWDW);
;  3979     3970  3      IF .WINDOW_LIST[1] NEQ 0
;  3980     3971  3          THEN BBLOCK[.WINDOW_LIST[1], WDW_LINK] = .DYNWDW;
;  3981     3972  3      IF .WINDOW_LIST[0] EQL 0 THEN WINDOW_LIST[0] = .DYNWDW;
```

```
; 3982      3973 2     END;
; 3983      3974 2
; 3984      3975 2
; 3985      3976 2  ! Return a pointer to the first window block (if any).
; 3986      3977 2  !
; 3987      3978 2  .WINDOW_LIST[0]
; 3988      3979 1 END;


                                     032AD          .BLKB    3
                            00000088 032B0 P.ABM:   .LONG    136                                          ;


                                OFFC 00000 CREATE_WINDOW:
                                                    .WORD    Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11        ; 3745
                 5E      FD58   CE  9E 00002        MOVAB    -680(SP), SP                                 ;
                 5A        04   AC  D0 00007        MOVL     P_HEADER, HEADER                             ; 3792
          04     AE        08   AC  D0 0000B        MOVL     P_RVN, RVN                                   ; 3793
                         0098   CE  7C 00010        CLRQ     WINDOW_LIST                                  ; 3794
                           10   AE  7C 00014        CLRQ     WINDOW                                       ; 3795
                 59        10   AE  9E 00017        MOVAB    WINDOW, P                                    ; 3797
                 50        01   AA  9E 0001B 1$:    MOVAB    1(HEADER), R0                                ; 3818
                 02 00000000'   EF  D1 0001F        CMPL     STRUCTURE_LEVEL, #2                          ; 3815
                           14   12 00026            BNEQ     2$                                           ;
                 50        60   9A 00028            MOVZBL   (R0), R0                                     ; 3818
                 5B      6A40   3E 0002B            MOVAW    (HEADER)[R0], R11                            ;
                 58        5B   D0 0002F            MOVL     R11, MAP_POINTER                             ;
                 50        3A   AA  9A 00032        MOVZBL   58(HEADER), R0                               ; 3819
                 6E      6840   3E 00036            MOVAW    (MAP_POINTER)[R0], END_MAP                   ;
                           16   11 0003A            BRB      3$                                           ; 3815
                 50        60   9A 0003C 2$:        MOVZBL   (R0), R0                                     ; 3823
                 5B      6A40   3E 0003F            MOVAW    (HEADER)[R0], R11                            ;
                 58        5B   D0 00043            MOVL     R11, MAP_POINTER                             ;
                 50        08   A8  9A 00046        MOVZBL   8(MAP_POINTER), R0                           ; 3824
                 6E   0A A840   3E 0004A            MOVAW    10(MAP_POINTER)[R0], END_MAP                 ;
                 58        0A   C0 0004F            ADDL2    #10, MAP_POINTER                             ; 3825
                 6E        58   D1 00052 3$:        CMPL     MAP_POINTER, END_MAP                         ; 3831
                           03   1F 00055            BLSSU    4$                                           ;
                         0093   31 00057            BRW      13$                                          ;
                 02 00000000'   EF  D1 0005A 4$:    CMPL     STRUCTURE_LEVEL, #2                          ; 3836
                           08   12 00061            BNEQ     5$                                           ;
                    00000000G   EF  16 00063        JSB      GET_MAP_POINTER                              ; 3838
                           0F   11 00069            BRB      6$                                           ;
                 57        02   A8  3C 0006B 5$:    MOVZWL   2(MAP_POINTER), LBN                          ; 3841
   57       08   10        88   F0 0006F            INSV     (MAP_POINTER)+, #16, #8, LBN                 ; 3842
                 56        FD   A8  9A 00074        MOVZBL   -3(MAP_POINTER), COUNT                       ; 3843
                           56   D6 00078            INCL     COUNT                                        ;
                           14   AE  D5 0007A 6$:    TSTL     WINDOW+4                                     ; 3853
                           0F   13 0007D            BEQL     8$                                           ;
            50   69        04   A9  C1 0007F        ADDL3    4(P), (P), R0                                ; 3855
                 57        50   D1 00084            CMPL     R0, LBN                                      ;
                           05   12 00087            BNEQ     8$                                           ;
                 69        56   C0 00089            ADDL2    COUNT, (P)                                   ; 3860
                           C4   11 0008C 7$:        BRB      3$                                           ;
                 10        14   AE  D1 0008E 8$:    CMPL     WINDOW+4, #16                                ; 3863
```

```
                                        4E  1F 00092          BLSSU   12$                                     ;
                                    08  AE  9F 00094          PUSHAB  DYNWDW                                  ; 3874
                                FF61    CF  9F 00097          PUSHAB  P,ABM                                   ; 3875
                00000000G   00          02  FB 0009B          CALLS   #2, LIB$GET_VM                          ;
                            11          50  E8 000A2          BLBS    STATUS, 9$                              ; 3877
                                        50  DD 000A5          PUSHL   STATUS
                                        7E  D4 000A7          CLRL    -(SP)
                                00000000G   8F  DD 000A9          PUSHL   #VERIFY$_ALLOCMEM
                00000000G   00          03  FB 000AF          CALLS   #3, LIB$SIGNAL                          ;
    08  BE          10  AE      0088    8F  28 00086  9$:     MOVC3   #136, WINDOW, @DYNWDW                   ; 3881
                            50      009C    CE  D0 000BE          MOVL    WINDOW_LIST+4, R0                       ; 3882
                                        04  13 000C3          BEQL    10$                                     ;
                            60          08  AE  D0 000C5          MOVL    DYNWDW, (R0)                            ; 3883
                                    0098    CE  D5 000C9  10$:     TSTL    WINDOW_LIST                             ; 3884
                                        06  12 000CD          BNEQ    11$                                     ;
                        0098    CE          08  AE  D0 000CF          MOVL    DYNWDW, WINDOW_LIST                     ;
                        009C    CE          08  AE  D0 000D5  11$:     MOVL    DYNWDW, WINDOW_LIST+4                   ; 3885
                                            10  AE  7C 000DB          CLRQ    WINDOW                                  ; 3886
                            59              10  AE  9E 000DE          MOVAB   WINDOW, P                               ; 3888
                                            14  AE  D6 000E2  12$:     INCL    WINDOW+4                                ; 3894
                            59                  08  C0 000E5          ADDL2   #8, P                                   ; 3895
                            69                      56  7D 000E8          MOVQ    COUNT, (P)                              ; 3896
                                                    9F  11 000EB          BRB     7$                                      ; 3831
                                                    50  D4 000ED  13$:     CLRL    R0                                      ; 3906
                            02 00000000'            EF  D1 000EF          CMPL    STRUCTURE_LEVEL, #2                     ;
                                                    07  12 000F6          BNEQ    14$                                     ;
                                                    50  D6 000F8          INCL    R0                                      ;
                                                0E  AA  B5 000FA          TSTW    14(HEADER)                              ; 3908
                                                    06  11 000FD          BRB     15$                                     ;
                                58                  5B  D0 000FF  14$:     MOVL    R11, MAP_POINTER                        ; 3911
                                                02  A8  B5 00102          TSTW    2(MAP_POINTER)                          ; 3912
                                                    3C  13 00105  15$:     BEQL    19$                                     ;
                                            0C          50  E9 00107          BLBC    R0, 16$                                 ; 3921
                                        F8  AD      0E  AA  D0 0010A          MOVL    14(HEADER), EXT_FILE_ID                 ; 3924
                                        FC  AD      12  AA  B0 0010F          MOVW    18(HEADER), EXT_FILE_ID+4               ; 3926
                                                    09  11 00114          BRB     17$                                     ; 3921
                                        F8  AD      02  A8  D0 00116  16$:     MOVL    2(MAP_POINTER), EXT_FILE_ID             ; 3930
                                        FC  AD          01  B0 0011B          MOVW    #1, EXT_FILE_ID+4                       ; 3932
                                                FC  AD  95 0011F  17$:     TSTB    EXT_FILE_ID+4                           ; 3934
                                                    05  12 00122          BNEQ    18$                                     ;
                                        FC  AD      04  AE  90 00124          MOVB    RVN, EXT_FILE_ID+4                      ;
                                            5A  00A0    CE  9E 00129  18$:     MOVAB   LOCAL_HEADER, HEADER                    ; 3939
                                    04  AE          FC  AD  9A 0012E          MOVZBL  EXT_FILE_ID+4, RVN                      ; 3940
                                            5A          DD 00133          PUSHL   HEADER                                  ; 3946
                                                F8  AD  9F 00135          PUSHAB  EXT_FILE_ID                             ;
                                0000V       CF          02  FB 00138          CALLS   #2, READ_HEADER                         ;
                                            03          50  E9 0013D          BLBC    R0, 19$                                 ;
                                                    FED8    31 00140          BRW     1$                                      ;
                                            52      14  AE  D0 00143  19$:     MOVL    WINDOW+4, R2                            ; 3954
                                                        48  13 00147          BEQL    22$                                     ;
                                                    0C  AE  9F 00149          PUSHAB  DYNWDW                                  ; 3962
                                            52          08  C4 0014C          MULL2   #8, R2                                  ; 3963
                                            52          08  C0 0014F          ADDL2   #8, R2                                  ;
                                        08  AE          52  D0 00152          MOVL    R2, 8(SP)                               ;
                                                08  AE  9F 00156          PUSHAB  8(SP)                                   ;
                            00000000G   00          02  FB 00159          CALLS   #2, LIB$GET_VM                          ;
                                        11          50  E8 00160          BLBS    STATUS, 20$                             ; 3965
```

```
                                                50  DD 00163          PUSHL   STATUS                                    ;
                                                7E  D4 00165          CLRL    -(SP)                                   ;
                                    00000000G   8F  DD 00167          PUSHL   #VERIFY$_ALLOCMEM                       ;
                          00000000G  00         03  FB 0016D          CALLS   #3, LIB$SIGNAL                          ;
                      0C  BE    10  AE         52  28 00174 20$:      MOVC3   R2, WINDOW, @DYNWDW                     ;3969
                                    50    009C  CE  D0 0017A          MOVL    WINDOW_LIST+4, R0                       ;3970
                                                04  13 0017F          BEQL    21$                                     ;
                                    60      0C  AE  D0 00181          MOVL    DYNWDW, (R0)                            ;3971
                                          0098  CE  D5 00185 21$:      TSTL    WINDOW_LIST                             ;3972
                                                06  12 00189          BNEQ    22$                                     ;
                                0098    CE      0C  AE  D0 0018B      MOVL    DYNWDW, WINDOW_LIST                     ;
                                    50    0098  CE  D0 00191 22$:      MOVL    WINDOW_LIST, R0                         ;3978
                                                04  00196            RET                                             ;3979

; Routine Size:  407 bytes,    Routine Base:  CODE + 3284
```

```
;  3990     3980  1 ROUTINE DELETE_WINDOW(WINDOW): NOVALUE=
;  3991     3981  1
;  3992     3982  1 !++
;  3993     3983  1 !
;  3994     3984  1 ! FUNCTIONAL DESCRIPTION:
;  3995     3985  1 !       This routine deletes a window block (or blocks).
;  3996     3986  1 !
;  3997     3987  1 ! INPUT PARAMETERS:
;  3998     3988  1 !       WINDOW          - Pointer to window block.
;  3999     3989  1 !
;  4000     3990  1 ! IMPLICIT INPUTS:
;  4001     3991  1 !       NONE
;  4002     3992  1 !
;  4003     3993  1 ! OUTPUT PARAMETERS:
;  4004     3994  1 !       NONE
;  4005     3995  1 !
;  4006     3996  1 ! IMPLICIT OUTPUTS:
;  4007     3997  1 !       NONE
;  4008     3998  1 !
;  4009     3999  1 ! ROUTINE VALUE:
;  4010     4000  1 !       NONE
;  4011     4001  1 !
;  4012     4002  1 ! SIDE EFFECTS:
;  4013     4003  1 !       Window blocks released.
;  4014     4004  1 !
;  4015     4005  1 !--
;  4016     4006  1
;  4017     4007  2 BEGIN
;  4018     4008  2 MAP
;  4019     4009  2         WINDOW:         REF BBLOCK;     ! Pointer to window block
;  4020     4010  2 LOCAL
;  4021     4011  2         W:              REF BBLOCK;     ! Pointer to window block
;  4022     4012  2
;  4023     4013  2
;  4024     4014  2 W = .WINDOW;
;  4025     4015  2 WHILE .W NEQ 0 DO
;  4026     4016  2     BEGIN
;  4027     4017  2     LOCAL
;  4028     4018  2         NEXT:           REF BBLOCK,     ! Pointer to next window block
;  4029     4019  2         STATUS;                         ! Status return
;  4030     4020  2
;  4031     4021  2
;  4032     4022  2     NEXT = .W[WDW_LINK];                ! Point to next block
;  4033     4023  2     STATUS = LIB$FREE_VM(               ! Free current block
;  4034     4024  2         %REF(WDW_S_HEADER + .W[WDW_SIZE] * WDW_S_ENTRY),
;  4035     4025  2         W);
;  4036     4026  2     IF NOT .STATUS THEN SIGNAL(VERIFY$_FREEMEM, 0, .STATUS);
;  4037     4027  2     W = .NEXT;                          ! Advance to next block
;  4038     4028  2     END;
;  4039     4029  1 END;


                                0004 00000 DELETE_WINDOW:
                                                .WORD   Save R2                 ; 3980
```

```
                 5E        08 C2 00002        SUBL2   #8, SP                        ;
          04     AE    04  AC D0 00005        MOVL    WINDOW, W                     ; 4014
                 50    04  AE D0 0000A 1$:    MOVL    W, R0                         ; 4015
                           37 13 0000E        BEQL    3$                            ;
                 52        60 D0 00010        MOVL    (R0), NEXT                    ; 4022
                       04  AE 9F 00013        PUSHAB  W                             ; 4023
                 50    04  A0 D0 00016        MOVL    4(R0), R0                     ; 4024
      04   AE    50        03 78 0001A        ASHL    #3, R0, 4(SP)                 ;
          04     AE        08 C0 0001F        ADDL2   #8, 4(SP)                     ;
                       04  AE 9F 00023        PUSHAB  4(SP)                         ;
   00000000G     00        02 FB 00026        CALLS   #2, LIB$FREE_VM               ;
                 11        50 E8 0002D        BLBS    STATUS, 2$                    ; 4026
                           50 DD 00030        PUSHL   STATUS                        ;
                           7E D4 00032        CLRL    -(SP)                         ;
                00000000G  8F DD 00034        PUSHL   #VERIFY$_FREEMEM              ;
   00000000G     00        03 FB 0003A        CALLS   #3, LIB$SIGNAL                ;
          04     AE        52 D0 00041 2$:    MOVL    NEXT, W                       ; 4027
                           C3 11 00045        BRB     1$                            ; 4015
                              04 00047 3$:    RET                                   ; 4029

; Routine Size:  72 bytes,    Routine Base:  CODE + 344B
```

```
;  4041     4030  1 ROUTINE MAP_VIRTUAL(WINDOW,VBN,LBN)=
;  4042     4031  1
;  4043     4032  1 !++
;  4044     4033  1 !
;  4045     4034  1 ! FUNCTIONAL DESCRIPTION:
;  4046     4035  1 !       This routine maps a virtual block number to a logical block number
;  4047     4036  1 !       using the specified window.
;  4048     4037  1 !
;  4049     4038  1 ! INPUT PARAMETERS:
;  4050     4039  1 !       WINDOW          - Pointer to a window block.
;  4051     4040  1 !       VBN             - Virtual block number.
;  4052     4041  1 !       LBN             - Pointer to where logical block number is returned.
;  4053     4042  1 !
;  4054     4043  1 ! IMPLICIT INPUTS:
;  4055     4044  1 !       NONE
;  4056     4045  1 !
;  4057     4046  1 ! OUTPUT PARAMETERS:
;  4058     4047  1 !       NONE
;  4059     4048  1 !
;  4060     4049  1 ! IMPLICIT OUTPUTS:
;  4061     4050  1 !       NONE
;  4062     4051  1 !
;  4063     4052  1 ! ROUTINE VALUE:
;  4064     4053  1 !       SS$_NORMAL if translation is successful or SS$_ENDOFFILE if the
;  4065     4054  1 !       specified virtual block is not within the file.
;  4066     4055  1 !
;  4067     4056  1 ! SIDE EFFECTS:
;  4068     4057  1 !       NONE
;  4069     4058  1 !
;  4070     4059  1 !--
;  4071     4060  1
;  4072     4061  2 BEGIN
;  4073     4062  2 LOCAL
;  4074     4063  2         W:              REF BBLOCK,     ! Pointer to window block
;  4075     4064  2         P:              REF BBLOCK,     ! Pointer to window block entry
;  4076     4065  2         N;                              ! VBN mapped so far
;  4077     4066  2
;  4078     4067  2
;  4079     4068  2 ! The virtual block number must not be 0.
;  4080     4069  2 !
;  4081     4070  2 IF .VBN EQL 0
;  4082     4071  2 THEN
;  4083     4072  2     RETURN SS$_ENDOFFILE;
;  4084     4073  2
;  4085     4074  2
;  4086     4075  2 ! Loop over the window blocks.
;  4087     4076  2 !
;  4088     4077  2 W = .WINDOW;
;  4089     4078  2 N = 1;
;  4090     4079  2 WHILE .W NEQ 0 DO
;  4091     4080  3     BEGIN
;  4092     4081  3     P = .W + WDW_S_HEADER;
;  4093     4082  3
;  4094     4083  3
;  4095     4084  3     ! Loop over the entries within the window block.
;  4096     4085  3     !
;  4097     4086  3     DECR I FROM .W[WDW_SIZE] TO 1 DO
```

```
;  4098     4087  4          BEGIN
;  4099     4088  4
;  4100     4089  4          ! If this entry maps the specified VBN, compute the LBN and return.
;  4101     4090  4          !
;  4102     4091  4          IF .VBN GEQU .N AND .VBN LSSU .N + .P[WDW_COUNT]
;  4103     4092  4          THEN
;  4104     4093  5              BEGIN
;  4105     4094  5              .LBN = .P[WDW_LBN] + .VBN - .N;
;  4106     4095  5              RETURN SS$_NORMAL;
;  4107     4096  4              END;
;  4108     4097  4
;  4109     4098  4
;  4110     4099  4          ! Advance to next entry.
;  4111     4100  4          !
;  4112     4101  4          N = .N + .P[WDW_COUNT];
;  4113     4102  4          P = .P + WDW_S_ENTRY;
;  4114     4103  3          END;
;  4115     4104  3
;  4116     4105  3
;  4117     4106  3      ! Advance to next window block.
;  4118     4107  3      !
;  4119     4108  3      W = .W[WDW_LINK];
;  4120     4109  2      END;
;  4121     4110  2
;  4122     4111  2
;  4123     4112  2 ! There were not enough mapping pointers to advance to the specified virtual
;  4124     4113  2 ! block number.  Therefore, return SS$_ENDOFFILE.
;  4125     4114  2 !
;  4126     4115  2 SS$_ENDOFFILE
;  4127     4116  1 END;


                                003C 00000 MAP_VIRTUAL:
                                           .WORD   Save R2,R3,R4,R5                ; 4030
                        55      08   AC  D0 00002          MOVL    VBN, R5         ; 4070
                                     40  13 00006          BEQL    5$              ;
                        50      04   AC  D0 00008          MOVL    WINDOW, W       ; 4077
                        52           01  D0 0000C          MOVL    #1, N           ; 4078
                                     50  D5 0000F 1$:      TSTL    W               ; 4079
                                     35  13 00011          BEQL    5$              ;
                        51      08   A0  9E 00013          MOVAB   8(R0), P        ; 4081
                54      04   A0          01  C1 00017      ADDL3   #1, 4(W), I     ; 4086
                                         22  11 0001C      BRB     4$              ;
                        52               55  D1 0001E 2$:  CMPL    R5, N           ; 4091
                                         17  1F 00021      BLSSU   3$              ;
                53      52               61  C1 00023      ADDL3   (P), N, R3      ;
                        53               55  D1 00027      CMPL    R5, R3          ;
                                         0E  1E 0002A      BGEQU   3$              ;
                53      55       04      A1  C1 0002C      ADDL3   4(P), R5, R3    ; 4094
            0C  BC      53               52  C3 00031      SUBL3   N, R3, @LBN     ;
                        50               01  D0 00036      MOVL    #1, R0          ; 4095
                                         04  00039      RET                     ;
                        52               81  C0 0003A 3$:  ADDL2   (P)+, N        ; 4101
                        51               04  C0 0003D      ADDL2   #4, P           ; 4102
```

```
                DB              54  F5 00040 4$:    SOBGTR  I, 2$                   ; 4086
                50              60  D0 00043      MOVL    (W), W                  ; 4108
                                C7  11 00046      BRB     1$                      ; 4079
                50      0870    8F  3C 00048 5$:  MOVZWL  #2160, R0               ; 4116
                                    04 0004D      RET                             ;

; Routine Size:  78 bytes,    Routine Base:  CODE + 3493
```

```
;  4129     4117  1 ROUTINE ACCESS_INDEX_2(RVN): NOVALUE=
;  4130     4118  1
;  4131     4119  1 !++
;  4132     4120  1 !
;  4133     4121  1 ! FUNCTIONAL DESCRIPTION:
;  4134     4122  1 !       This routine accesses an index file on the second channel.  If the
;  4135     4123  1 !       second channel had an index file accessed, it is first deaccessed.
;  4136     4124  1 !
;  4137     4125  1 ! INPUT PARAMETERS:
;  4138     4126  1 !       RVN             - Relative volume number.  If zero, no new index file
;  4139     4127  1 !                         is accessed, but a previous one is deaccessed.
;  4140     4128  1 !
;  4141     4129  1 ! IMPLICIT INPUTS:
;  4142     4130  1 !       CHAN2_RVN       - RVN on which index file is currently accessed on
;  4143     4131  1 !                         the second channel.  If zero, none is currently
;  4144     4132  1 !                         accessed.
;  4145     4133  1 !
;  4146     4134  1 ! OUTPUT PARAMETERS:
;  4147     4135  1 !       NONE
;  4148     4136  1 !
;  4149     4137  1 ! IMPLICIT OUTPUTS:
;  4150     4138  1 !       CHAN2_RVN       - Updated to equal RVN.
;  4151     4139  1 !
;  4152     4140  1 ! ROUTINE VALUE:
;  4153     4141  1 !       NONE
;  4154     4142  1 !
;  4155     4143  1 ! SIDE EFFECTS:
;  4156     4144  1 !       If RVN is nonzero, index file on specified RVN accessed on second
;  4157     4145  1 !       channel.
;  4158     4146  1 !
;  4159     4147  1 !--
;  4160     4148  1
;  4161     4149  2 BEGIN
;  4162     4150  2 LOCAL
;  4163     4151  2         STATUS;                 ! Status variable
;  4164     4152  2
;  4165     4153  2
;  4166     4154  2 IF .RVN NEQ .CHAN2_RVN
;  4167     4155  2 THEN
;  4168     4156  3     BEGIN
;  4169     4157  3
;  4170     4158  3     ! If an index file is currently accessed on the second channel,
;  4171     4159  3     ! deaccess it.
;  4172     4160  3     !
;  4173     4161  3     IF .CHAN2_RVN NEQ 0
;  4174     4162  3     THEN
;  4175   P 4163  3         $QIOW(
;  4176   P 4164  3             FUNC=IO$_DEACCESS,
;  4177     4165  3             CHAN=.CHANNEL_2);
;  4178     4166  3
;  4179     4167  3
;  4180     4168  3     ! If a new index file is to be accessed, access it.
;  4181     4169  3     !
;  4182     4170  3     IF .RVN NEQ 0
;  4183     4171  3     THEN
;  4184     4172  4         BEGIN
;  4185     4173  4         CH$FILL(0, FIB$C_LENGTH, FIB);
```

```
: 4186          4174  4          FIB[FIB$L_ACCTL] = .ACCTL[.RVN-1];
: 4187          4175  4          FIB[FIB$W_FID_NUM] = FID$C_INDEXF;
: 4188          4176  4          FIB[FIB$W_FID_SEQ] = FID$C_INDEXF;
: 4189          4177  4          FIB[FIB$W_FID_RVN] = .RVN;
: 4190        P 4178  4          STATUS = $QIOW(
: 4191        P 4179  4              FUNC=IO$_ACCESS OR IO$M_ACCESS,
: 4192        P 4180  4              CHAN=.CHANNEL_2,
: 4193        P 4181  4              IOSB=IOSB,
: 4194          4182  4              P1=FIB_DESC);
: 4195          4183  4          IF .STATUS THEN STATUS = .IOSB[0];
: 4196          4184  4          IF NOT .STATUS THEN SIGNAL(VERIFY$_OPENINDEX, 1, .RVN, .STATUS);
: 4197          4185  4          END;
: 4198          4186
: 4199          4187
: 4200          4188      ! Update the status variable.
: 4201          4189      !
: 4202          4190  3   CHAN2_RVN = .RVN;
: 4203          4191  2   END;
: 4204          4192  1 END;


                                01FC 00000 ACCESS_INDEX_2:
                                           .WORD   Save R2,R3,R4,R5,R6,R7,R8                    ; 4117
                58 00000000G   00  9E 00002         MOVAB   SYS$QIOW, R8
                57 00000000'   EF  9E 00009         MOVAB   CHAN2_RVN, R7
                56       04    AC  D0 00010         MOVL    RVN, R6                              ; 4154
                67             56  D1 00014         CMPL    R6, CHAN2_RVN
                               01  12 00017         BNEQ    1$
                                   04 00019         RET
                               67  D5 0001A 1$:      TSTL    CHAN2_RVN                            ; 4161
                               13  13 0001C         BEQL    2$
                               7E  7C 0001E         CLRQ    -(SP)                                ; 4165
                               7E  7C 00020         CLRQ    -(SP)
                               7E  7C 00022         CLRQ    -(SP)
                               7E  7C 00024         CLRQ    -(SP)
                          7E       34 7D 00026         MOVQ    #52, -(SP)
                       FC  A7      DD 00029         PUSHL   CHANNEL_2
                               7E  D4 0002C         CLRL    -(SP)
                          68       0C FB 0002E         CALLS   #12, SYS$QIOW
                               56  D5 00031 2$:      TSTL    R6                                   ; 4170
                               6C  13 00033         BEQL    4$
0040   BF        00            6E  00 2C 00035         MOVC5   #0, (SP), #0, #64, FIB               ; 4173
                       00000000' EF    0003C
                50 00000000'FF46  DE 00041         MOVAL   @ACCTL[R6], R0                       ; 4174
          00000000' EF       FC  A0 D0 00049         MOVL    -4(R0), FIB
          00000000' EF 00010001  8F D0 00051         MOVL    #65537, FIB+4                        ; 4175
          00000000' EF           56 B0 0005C         MOVW    R6, FIB+8                            ; 4177
                               7E  7C 00063         CLRQ    -(SP)                                ; 4182
                               7E  7C 00065         CLRQ    -(SP)
                               7E  D4 00067         CLRL    -(SP)
                         CAE2  CF  9F 00069         PUSHAB  FIB_DESC
                               7E  7C 0006D         CLRQ    -(SP)
                       00000000' EF  9F 0006F         PUSHAB  IOSB
                          7E       72  8F 9A 00075         MOVZBL  #114, -(SP)
```

```
                                 FC  A7  DD 00079         PUSHL   CHANNEL_2                                    ;
                                     7E  D4 0007C         CLRL    -(SP)                                       ;
                                 68  0C  FB 0007E         CALLS   #12, SYS$QIOW                               ;
                                 0A  50  E9 00081         BLBC    STATUS, 3$                                  ; 4183
                                 50  00000000'  EF  3C 00084  MOVZWL  IOSB, STATUS                            ;
                                 13  50  E8 0008B         BLBS    STATUS, 4$                                  ; 4184
                                     50  DD 0008E  3$:     PUSHL   STATUS                                      ;
                                     56  DD 00090         PUSHL   R6                                          ;
                                     01  DD 00092         PUSHL   #1                                          ;
                                 00000000G  8F  DD 00094  PUSHL   #VERIFY$_OPENINDEX                          ;
                         00000000G  00  04  FB 0009A      CALLS   #4, LIB$SIGNAL                              ;
                                 67  56  D0 000A1  4$:     MOVL    R6, CHAN2_RVN                               ; 4190
                                     04  000A4             RET                                                 ; 4192

; Routine Size:  165 bytes,    Routine Base:  CODE + 34E1
```

```
;  4206     4193  1 ROUTINE COUNT_QUOTA(UIC,QBLOCKS,IBLOCKS): NOVALUE=
;  4207     4194  1
;  4208     4195  1 !++
;  4209     4196  1 !
;  4210     4197  1 ! FUNCTIONAL DESCRIPTION:
;  4211     4198  1 !       This routine maintains the quota data base.
;  4212     4199  1 !
;  4213     4200  1 ! INPUT PARAMETERS:
;  4214     4201  1 !       UIC             - UIC to be updated.
;  4215     4202  1 !       QBLOCKS         - Blocks to be added to amount used (per quota file)
;  4216     4203  1 !       IBLOCKS         - Blocks to be added to amount used (per index file)
;  4217     4204  1 !
;  4218     4205  1 ! IMPLICIT INPUTS:
;  4219     4206  1 !       NONE
;  4220     4207  1 !
;  4221     4208  1 ! OUTPUT PARAMETERS:
;  4222     4209  1 !       NONE
;  4223     4210  1 !
;  4224     4211  1 ! IMPLICIT OUTPUTS:
;  4225     4212  1 !       NONE
;  4226     4213  1 !
;  4227     4214  1 ! ROUTINE VALUE:
;  4228     4215  1 !       NONE
;  4229     4216  1 !
;  4230     4217  1 ! SIDE EFFECTS:
;  4231     4218  1 !       NONE
;  4232     4219  1 !
;  4233     4220  1 !--
;  4234     4221  1
;  4235     4222  2 BEGIN
;  4236     4223  2 LOCAL
;  4237     4224  2         STATUS,                         ! Status variable
;  4238     4225  2         T:              REF BBLOCK,     ! Pointer to table segment
;  4239     4226  2         E:              REF BBLOCK,     ! Pointer to table entry
;  4240     4227  2         J;                              ! Index of table entry
;  4241     4228  2
;  4242     4229  2
;  4243     4230  2 T = QT;
;  4244     4231  2 J = -1;
;  4245     4232  2 CASE
;  4246     4233  3     BEGIN
;  4247     4234  3     WHILE 1 DO
;  4248     4235  4         BEGIN
;  4249     4236  4         IF .J GTRU .T[QT_COUNT]                         ! More entries?
;  4250     4237  4         THEN                                            ! No more entries
;  4251     4238  4             IF .T[QT_LINK] EQL 0                        ! More blocks?
;  4252     4239  4             THEN                                        ! No more blocks
;  4253     4240  4                 IF .T[QT_COUNT] EQL QT_MAXCOUNT         ! More space?
;  4254     4241  4                 THEN                                    ! No more space
;  4255     4242  4                     EXITLOOP 1                          ! Add new block
;  4256     4243  4                 ELSE                                    ! More space
;  4257     4244  4                     EXITLOOP -1                         ! Use existing space
;  4258     4245  4             ELSE                                        ! More blocks
;  4259     4246  5                 BEGIN
;  4260     4247  5                 T = .T[QT_LINK];                        ! Point to next block
;  4261     4248  5                 E = .T + QT_S_HDR;                      ! Point to first entry
;  4262     4249  5                 J = 1;                                  ! Count first entry
```

```
;  4263     4250  5                END
;  4264     4251  4            ELSE
;  4265     4252  4                IF .E[QT_UIC] EQL .UIC                  ! More entries
;  4266     4253  4                THEN                                    ! Correct entry?
;  4267     4254  4                    EXITLOOP 0                          ! Update existing entry
;  4268     4255  4                ELSE
;  4269     4256  5                    BEGIN
;  4270     4257  5                    E = .E + QT_S_ENT;                  ! Point to next entry
;  4271     4258  5                    J = .J + 1;                         ! Count next entry
;  4272     4259  5                    END
;  4273     4260  4            END
;  4274     4261  3        END
;  4275     4262  2    FROM -1 TO 1 OF
;  4276     4263  2        SET
;  4277     4264  2
;  4278     4265  2        [-1]:
;  4279     4266  3            BEGIN
;  4280     4267  3            !
;  4281     4268  3            ! Add new entry.  E points to space where it can be added, T points
;  4282     4269  3            ! to table segment.
;  4283     4270  3            !
;  4284     4271  3            T[QT_COUNT] = .T[QT_COUNT] + 1;
;  4285     4272  3            E[QT_UIC] = .UIC;
;  4286     4273  3            E[QT_QUO_USED] = .QBLOCKS;
;  4287     4274  3            E[QT_IDX_USED] = .IBLOCKS;
;  4288     4275  3            END;
;  4289     4276  2
;  4290     4277  2        [0]:
;  4291     4278  3            BEGIN
;  4292     4279  3            !
;  4293     4280  3            ! Update existing entry.  E points to the entry.
;  4294     4281  3            !
;  4295     4282  3            E[QT_QUO_USED] = .E[QT_QUO_USED] + .QBLOCKS;
;  4296     4283  3            E[QT_IDX_USED] = .E[QT_IDX_USED] + .IBLOCKS;
;  4297     4284  3            END;
;  4298     4285  2
;  4299     4286  2        [1]:
;  4300     4287  3            BEGIN
;  4301     4288  3            !
;  4302     4289  3            ! Add new table segment.  T points to existing table segment.
;  4303     4290  3            !
;  4304     4291  3            STATUS = LIB$GET_VM(
;  4305     4292  3                UPLIT(QT_S_HDR + QT_MAXCOUNT*QT_S_ENT),
;  4306     4293  3                T[QT_LINK]);
;  4307     4294  3            IF NOT .STATUS THEN SIGNAL(VERIFY$_ALLOCMEM, 0, .STATUS);
;  4308     4295  3            T = .T[QT_LINK];
;  4309     4296  3            T[QT_LINK] = 0;
;  4310     4297  3            T[QT_COUNT] = 1;
;  4311     4298  3            E = .T + QT_S_HDR;
;  4312     4299  3            E[QT_UIC] = .UIC;
;  4313     4300  3            E[QT_QUO_USED] = .QBLOCKS;
;  4314     4301  3            E[QT_IDX_USED] = .IBLOCKS;
;  4315     4302  3            END;
;  4316     4303  2
;  4317     4304  2        TES;
;  4318     4305  1    END;
; INFO#250              L1:4252
```

; Referenced LOCAL symbol E is probably not initialized

```
                                        03586                   .BLKB   2
                                00000C08 03588 P.ABN:           .LONG   3080                                    ;

                                         000C 00000 COUNT_QUOTA:
                                                                .WORD   Save R2,R3                              ; 4193
                    53 00000000'    EF   9E 00002               MOVAB   QT, T                               ; 4230
                    50              01   CE 00009               MNEGL   #1, J                               ; 4231
                04  A3              50   D1 0000C 1$:           CMPL    J, 4(T)                             ; 4236
                                    1F   1B 00010               BLEQU   3$                                  ;
                                    63   D5 00012               TSTL    (T)                                 ; 4238
                                    0F   12 00014               BNEQ    2$                                  ;
            00000100    8F      04  A3   D1 00016               CMPL    4(T), #256                          ; 4240
                                    22   12 0001E               BNEQ    5$                                  ;
                    50              01   D0 00020               MOVL    #1, R0                              ; 4242
                                    20   11 00023               BRB     6$                                  ;
                    53              63   D0 00025 2$:           MOVL    (T), T                              ; 4247
                    52          08  A3   9E 00028               MOVAB   8(R3), E                            ; 4248
                    50              01   D0 0002C               MOVL    #1, J                               ; 4249
                                    DB   11 0002F               BRB     1$                                  ; 4238
                04  AC              62   D1 00031 3$:           CMPL    (E), UIC                            ; 4252
                                    04   12 00035               BNEQ    4$                                  ;
                                    50   D4 00037               CLRL    R0                                  ; 4254
                                    0A   11 00039               BRB     6$                                  ;
                    52              0C   C0 0003B 4$:           ADDL2   #12, E                              ; 4257
                                    50   D6 0003E               INCL    J                                   ; 4258
                                    CA   11 00040               BRB     1$                                  ; 4235
                    50              01   CE 00042 5$:           MNEGL   #1, R0                              ; 4234
            02 FFFFFFFF     8F      50   CF 00045 6$:           CASEL   R0, #-1, #2                         ; 4233
0016                    000B            0006    0004D 7$:       .WORD   8$-7$,-                             ;
                                                                9$-7$,-                                     ;
                                                                10$-7$                                      ;
                                04  A3   D6 00053 8$:           INCL    4(T)                                ; 4271
                                    38   11 00056               BRB     12$                                 ; 4272
                04  A2          08  AC   C0 00058 9$:           ADDL2   QBLOCKS, 4(E)                       ; 4282
                08  A2          0C  AC   C0 0005D               ADDL2   IBLOCKS, 8(E)                       ; 4283
                                        04 00062                RET                                         ; 4232
                                    53   DD 00063 10$:          PUSHL   T                                   ; 4293
                                94  AF   9F 00065               PUSHAB  P.ABN                               ; 4292
            00000000G   00          02   FB 00068               CALLS   #2, LIB$GET_VM                      ; 4293
                        11          50   E8 0006F               BLBS    STATUS, 11$                         ; 4294
                                    50   DD 00072               PUSHL   STATUS                              ;
                                    7E   D4 00074               CLRL    -(SP)                               ;
                                00000000G 8F DD 00076           PUSHL   #VERIFY$_ALLOCMEM                   ;
            00000000G   00          03   FB 0007C               CALLS   #3, LIB$SIGNAL                      ;
                        53          63   D0 00083 11$:          MOVL    (T), T                              ; 4295
                                    63   D4 00086               CLRL    (T)                                 ; 4296
                04  A3              01   D0 00088               MOVL    #1, 4(T)                            ; 4297
                    52          08  A3   9E 0008C               MOVAB   8(R3), E                            ; 4298
                    62          04  AC   7D 00090 12$:          MOVQ    UIC, (E)                            ; 4299
                08  A2          0C  AC   D0 00094               MOVL    IBLOCKS, 8(E)                       ; 4301
                                        04 00099                RET                                         ; 4305
```

```
; Routine Size:  154 bytes,    Routine Base:  CODE + 358C
```

```
;  4320     4306  1 ROUTINE READ_HEADER(FILE_ID,BUFFER)=
;  4321     4307  1
;  4322     4308  1 !++
;  4323     4309  1 !
;  4324     4310  1 ! FUNCTIONAL DESCRIPTION:
;  4325     4311  1 !       This routine reads one file header into the specified buffer.
;  4326     4312  1 !
;  4327     4313  1 ! INPUT PARAMETERS:
;  4328     4314  1 !       FILE_ID         - File ID of header to be read.
;  4329     4315  1 !       BUFFER          - Pointer to buffer.
;  4330     4316  1 !
;  4331     4317  1 ! IMPLICIT INPUTS:
;  4332     4318  1 !       NONE
;  4333     4319  1 !
;  4334     4320  1 ! OUTPUT PARAMETERS:
;  4335     4321  1 !       NONE
;  4336     4322  1 !
;  4337     4323  1 ! IMPLICIT OUTPUTS:
;  4338     4324  1 !       NONE
;  4339     4325  1 !
;  4340     4326  1 ! ROUTINE VALUE:
;  4341     4327  1 !       True if header successfully read, false otherwise.
;  4342     4328  1 !
;  4343     4329  1 ! SIDE EFFECTS:
;  4344     4330  1 !       Header read into buffer.
;  4345     4331  1 !
;  4346     4332  1 !--
;  4347     4333  1
;  4348     4334  2 BEGIN
;  4349     4335  2 MAP
;  4350     4336  2         FILE_ID:        REF BBLOCK,     ! Pointer to file ID
;  4351     4337  2         BUFFER:         REF BBLOCK;     ! Pointer to header
;  4352     4338  2 LOCAL
;  4353     4339  2         STATUS,                         ! Status variable
;  4354     4340  2         FILE_NUMBER,                    ! Clean file number
;  4355     4341  2         RVN;                            ! Clean RVN
;  4356     4342  2
;  4357     4343  2
;  4358     4344  2 ! Get a clean file number and RVN and validity check.  If failure,
;  4359     4345  2 ! return failure.
;  4360     4346  2 !
;  4361     4347  2 FILE_NUMBER = .FILE_ID[FID$W_NUM];
;  4362     4348  2 FILE_NUMBER<16,8> = .FILE_ID[FID$B_NMX];
;  4363     4349  2 RVN = .FILE_ID[FID$B_RVN];
;  4364     4350  2 IF .RVN GTRU .VOLUME_COUNT THEN RETURN FALSE;
;  4365     4351  2 IF .FILE_NUMBER-1 GTRU .MAXFILIDX[.RVN-1] THEN RETURN FALSE;
;  4366     4352  2
;  4367     4353  2
;  4368     4354  2 ! Access index file on appropriate volume.
;  4369     4355  2 !
;  4370     4356  2 ACCESS_INDEX_2(.RVN);
;  4371     4357  2
;  4372     4358  2
;  4373     4359  2 ! Read the header.
;  4374     4360  2 !
;  4375   P 4361  2 STATUS = $QIOW(
;  4376   P 4362  2     FUNC=IO$_READVBLK,
```

```
;  4377        P 4363  2      CHAN=.CHANNEL_2,
;  4378        P 4364  2      IOSB=IOSB,
;  4379        P 4365  2      P1=.BUFFER,
;  4380        P 4366  2      P2=512,
;  4381          4367  2      P3=.FILE_NUMBER + .HEADER_OFFSET[.RVN-1]);
;  4382          4368  2  IF .STATUS THEN STATUS = .IOSB[0];
;  4383          4369  2
;  4384          4370  2
;  4385          4371  2  ! If failure, report it and return failure.
;  4386          4372  2  !
;  4387          4373  2  IF NOT .STATUS
;  4388          4374  2  THEN
;  4389          4375  3      BEGIN
;  4390          4376  3      HEADER_ERROR(VERIFY$_READHEADER, .FILE_ID, .BUFFER, .STATUS);
;  4391          4377  3      RETURN FALSE;
;  4392          4378  2      END;
;  4393          4379  2
;  4394          4380  2
;  4395          4381  2  ! Verify the header that was read.
;  4396          4382  2  !
;  4397          4383  2  STATUS = VERIFY_HEADER(.BUFFER, .FILE_ID);
;  4398          4384  2  IF NOT .STATUS
;  4399          4385  2  THEN
;  4400          4386  2      HEADER_ERROR(VERIFY$_BADHEADER, .FILE_ID, .BUFFER);
;  4401          4387  2
;  4402          4388  2
;  4403          4389  2  ! Return status of header.
;  4404          4390  2  !
;  4405          4391  2  .STATUS
;  4406          4392  1  END;


                                 003C 00000 READ_HEADER:
                                            .WORD   Save R2,R3,R4,R5                          ;  4306
                      55 00000000'   EF  9E 00002   MOVAB   IOSB, R5
                      54         04  AC  D0 00009   MOVL    FILE_ID, R4                       ;  4347
                      52             64  3C 0000D   MOVZWL  (R4), FILE_NUMBER
52                    08         10  05  A4  F0 00010   INSV    5(R4), #16, #8, FILE_NUMBER   ;  4348
                      53         04  A4  9A 00016   MOVZBL  4(R4), RVN                       ;  4349
                         08      A5      53  D1 0001A   CMPL    RVN, VOLUME_COUNT             ;  4350
                                         5F  1A 0001E   BGTRU   2$
                      51         FF  A2  9E 00020   MOVAB   -1(R2), R1                       ;  4351
                      50         64 B543  DE 00024   MOVAL   @MAXFILIDX[RVN], R0
                         FC      A0      51  D1 00029   CMPL    R1, -4(R0)
                                         50  1A 0002D   BGTRU   2$
                                         53  DD 0002F   PUSHL   RVN                           ;  4356
                      FE85       CF      01  FB 00031   CALLS   #1, ACCESS_INDEX_2
                                         7E  7C 00036   CLRQ    -(SP)                         ;  4367
                                         7E  D4 00038   CLRL    -(SP)
                      50       00A0 D543  DE 0003A   MOVAL   @HEADER_OFFSET[RVN], R0
                                  FC B042  9F 00040   PUSHAB  @-4(R0)[FILE_NUMBER]
                      7E       0200   8F  3C 00044   MOVZWL  #512, -(SP)
                                  08   AC  DD 00049   PUSHL   BUFFER
                                       7E  7C 0004C   CLRQ    -(SP)                         ;
```

```
                                 55  DD 0004E          PUSHL   R5                                          ;
                                 31  DD 00050          PUSHL   #49                                         ;
                      00000000'  EF  DD 00052          PUSHL   CHANNEL_2                                   ;
                                 7E  D4 00058          CLRL    -(SP)                                       ;
            00000000G  00        0C  FB 0005A          CALLS   #12, SYS$QIOW                               ;
                       53        50  D0 00061          MOVL    R0, STATUS                                  ;
                       06        53  E9 00064          BLBC    STATUS, 1$                                  ; 4368
                       53        65  3C 00067          MOVZWL  IOSB, STATUS                                ;
                       15        53  E8 0006A          BLBS    STATUS, 3$                                  ; 4373
                                 53  DD 0006D  1$:      PUSHL   STATUS                                      ; 4376
                             08  AC  DD 0006F          PUSHL   BUFFER                                      ;
                                 54  DD 00072          PUSHL   R4                                          ;
                      00000000G  8F  DD 00074          PUSHL   #VERIFY$_READHEADER                         ;
                0000V CF           04  FB 0007A          CALLS   #4, HEADER_ERROR                            ;
                                 50  D4 0007F  2$:      CLRL    R0                                          ; 4377
                                 04  00081             RET                                                 ;
                                 54  DD 00082  3$:      PUSHL   R4                                          ; 4383
                             08  AC  DD 00084          PUSHL   BUFFER                                      ;
                F948 CF           02  FB 00087          CALLS   #2, VERIFY_HEADER                           ;
                       53        50  D0 0008C          MOVL    R0, STATUS                                  ;
                       10        53  E8 0008F          BLBS    STATUS, 4$                                  ; 4384
                             08  AC  DD 00092          PUSHL   BUFFER                                      ; 4386
                                 54  DD 00095          PUSHL   R4                                          ;
                      00000000G  8F  DD 00097          PUSHL   #VERIFY$_BADHEADER                          ;
                0000V CF           03  FB 0009D          CALLS   #3, HEADER_ERROR                            ;
                       50        53  D0 000A2  4$:      MOVL    STATUS, R0                                  ; 4392
                                 04  000A5             RET                                                 ;

; Routine Size:  166 bytes,    Routine Base:  CODE + 3626
```

```
;  4408     4393  1 ROUTINE WRITE_HEADER(FILE_ID,BUFFER)=
;  4409     4394  1
;  4410     4395  1 !++
;  4411     4396  1 !
;  4412     4397  1 ! FUNCTIONAL DESCRIPTION:
;  4413     4398  1 !       This routine writes one file header from the specified buffer.
;  4414     4399  1 !
;  4415     4400  1 ! INPUT PARAMETERS:
;  4416     4401  1 !       FILE_ID         - File ID of header to be written.
;  4417     4402  1 !       BUFFER          - Pointer to buffer.
;  4418     4403  1 !
;  4419     4404  1 ! IMPLICIT INPUTS:
;  4420     4405  1 !       NONE
;  4421     4406  1 !
;  4422     4407  1 ! OUTPUT PARAMETERS:
;  4423     4408  1 !       NONE
;  4424     4409  1 !
;  4425     4410  1 ! IMPLICIT OUTPUTS:
;  4426     4411  1 !       NONE
;  4427     4412  1 !
;  4428     4413  1 ! ROUTINE VALUE:
;  4429     4414  1 !       Completion status.
;  4430     4415  1 !
;  4431     4416  1 ! SIDE EFFECTS:
;  4432     4417  1 !       Header written from buffer.
;  4433     4418  1 !
;  4434     4419  1 !--
;  4435     4420  1
;  4436     4421  2 BEGIN
;  4437     4422  2 MAP
;  4438     4423  2         FILE_ID:        REF BBLOCK,     ! Pointer to file ID
;  4439     4424  2         BUFFER:         REF BBLOCK;     ! Pointer to header
;  4440     4425  2 LOCAL
;  4441     4426  2         STATUS,                         ! Status variable
;  4442     4427  2         FILE_NUMBER,                    ! Clean file number
;  4443     4428  2         RVN;                            ! Clean RVN
;  4444     4429  2
;  4445     4430  2
;  4446     4431  2 ! Get a clean file number and RVN and validity check.  If failure,
;  4447     4432  2 ! do nothing.
;  4448     4433  2 !
;  4449     4434  2 FILE_NUMBER = .FILE_ID[FID$W_NUM];
;  4450     4435  2 FILE_NUMBER<16,8> = .FILE_ID[FID$B_NMX];
;  4451     4436  2 RVN = .FILE_ID[FID$B_RVN];
;  4452     4437  2 IF .RVN GTRU .VOLUME_COUNT THEN RETURN 0;
;  4453     4438  2 IF .FILE_NUMBER-1 GTRU .MAXFILIDX[.RVN-1] THEN RETURN 0;
;  4454     4439  2
;  4455     4440  2
;  4456     4441  2 ! Recompute the checksum.
;  4457     4442  2 !
;  4458     4443  2 CHECKSUM(.BUFFER);
;  4459     4444  2
;  4460     4445  2
;  4461     4446  2 ! Access the index file on the appropriate volume.
;  4462     4447  2 !
;  4463     4448  2 ACCESS_INDEX_2(.RVN);
;  4464     4449  2
```

```
;  4465          4450  2
;  4466          4451  2 ! Write the block.
;  4467          4452  2 !
;  4468        P 4453  2 STATUS = $QIOW(
;  4469        P 4454  2     FUNC=IO$_WRITEVBLK,
;  4470        P 4455  2     CHAN=.CHANNEL_2,
;  4471        P 4456  2     IOSB=IOSB,
;  4472        P 4457  2     P1=.BUFFER,
;  4473        P 4458  2     P2=512,
;  4474          4459  2     P3=.FILE_NUMBER + .HEADER_OFFSET[.RVN-1]);
;  4475          4460  2 IF .STATUS THEN STATUS = .IOSB[0];
;  4476          4461  2
;  4477          4462  2
;  4478          4463  2 ! If failure, report it.
;  4479          4464  2 !
;  4480          4465  2 IF NOT .STATUS
;  4481          4466  2 THEN
;  4482          4467  2     HEADER_ERROR(VERIFY$_WRITEHEADER, .FILE_ID, .BUFFER, .STATUS);
;  4483          4468  2
;  4484          4469  2
;  4485          4470  2 ! Return the completion status.
;  4486          4471  2 !
;  4487          4472  2 .STATUS
;  4488          4473  1 END;


                                    003C 00000 WRITE_HEADER:
                                               .WORD   Save R2,R3,R4,R5                ;  4393
                    55 00000000'  EF  9E 00002         MOVAB   IOSB, R5
                    54        04  AC  D0 00009         MOVL    FILE_ID, R4            ;  4434
                    52            64  3C 0000D         MOVZWL  (R4), FILE_NUMBER
52                  10        05  A4  F0 00010         INSV    5(R4), #16, #8, FILE_NUMBER  ;  4435
                    53        04  A4  9A 00016         MOVZBL  4(R4), RVN             ;  4436
                 08 A5            53  D1 0001A         CMPL    RVN, VOLUME_COUNT      ;  4437
                                  0F  1A 0001E         BGTRU   1$
                    51        FF  A2  9E 00020         MOVAB   -1(R2), R1             ;  4438
                    50     64 B543  DE 00024         MOVAL   @MAXFILIDX[RVN], R0
                 FC A0            51  D1 00029         CMPL    R1, -4(R0)
                                  03  1B 0002D         BLEQU   2$
                                  50  D4 0002F 1$:     CLRL    R0
                                      04 00031         RET
                                  08  AC  DD 00032 2$:  PUSHL   BUFFER                 ;  4443
             00000000G EF         01  FB 00035         CALLS   #1, CHECKSUM
                                  53  DD 0003C         PUSHL   RVN                    ;  4448
                FDD2 CF           01  FB 0003E         CALLS   #1, ACCESS_INDEX_2
                                  7E  7C 00043         CLRQ    -(SP)                  ;  4459
                                  7E  D4 00045         CLRL    -(SP)
                    50     00A0 D543  DE 00047         MOVAL   @HEADER_OFFSET[RVN], R0
                              FC B042  9F 0004D         PUSHAB  @-4(R0)[FILE_NUMBER]
                    7E       0200  8F  3C 00051         MOVZWL  #512, -(SP)
                                  08  AC  DD 00056     PUSHL   BUFFER
                                  7E  7C 00059         CLRQ    -(SP)
                                  55  DD 0005B         PUSHL   R5
                                  30  DD 0005D         PUSHL   #48
```

```
                          00000000'  EF  DD 0005F                PUSHL   CHANNEL_2                                ;
                                     7E  D4 00065                CLRL    -(SP)                                    ;
                 00000000G  00       0C  FB 00067                CALLS   #12, SYS$QIOW                            ;
                            53       50  D0 0006E                MOVL    R0, STATUS                               ;
                            06       53  E9 00071                BLBC    STATUS, 3$                               ; 4460
                            53       65  3C 00074                MOVZWL  IOSB, STATUS                             ;
                            12       53  E8 00077                BLBS    STATUS, 4$                               ; 4465
                                     53  DD 0007A 3$:            PUSHL   STATUS                                   ; 4467
                                08   AC  DD 0007C                PUSHL   BUFFER                                   ;
                                     54  DD 0007F                PUSHL   R4                                       ;
                          00000000G  8F  DD 00081                PUSHL   #VERIFY$_WRITEHEADER                     ;
                      0000V  CF       04  FB 00087                CALLS   #4, HEADER_ERROR                         ;
                            50       53  D0 0008C 4$:            MOVL    STATUS, R0                               ; 4473
                                     04  0008F                   RET                                              ;

; Routine Size:  144 bytes,    Routine Base:  CODE + 36CC
```

```
;  4490     4474  1 ROUTINE DELETE_HEADER(FILE_ID,HEADER): NOVALUE=
;  4491     4475  1
;  4492     4476  1 !++
;  4493     4477  1 !
;  4494     4478  1 ! FUNCTIONAL DESCRIPTION:
;  4495     4479  1 !       This routine writes a deleted file header.
;  4496     4480  1 !
;  4497     4481  1 ! INPUT PARAMETERS:
;  4498     4482  1 !       FILE_ID         - File ID of the header.
;  4499     4483  1 !       HEADER          - Pointer to file header.
;  4500     4484  1 !
;  4501     4485  1 ! IMPLICIT INPUTS:
;  4502     4486  1 !       NONE
;  4503     4487  1 !
;  4504     4488  1 ! OUTPUT PARAMETERS:
;  4505     4489  1 !       NONE
;  4506     4490  1 !
;  4507     4491  1 ! IMPLICIT OUTPUTS:
;  4508     4492  1 !       NONE
;  4509     4493  1 !
;  4510     4494  1 ! ROUTINE VALUE:
;  4511     4495  1 !       NONE
;  4512     4496  1 !
;  4513     4497  1 ! SIDE EFFECTS:
;  4514     4498  1 !       File header written.
;  4515     4499  1 !
;  4516     4500  1 !--
;  4517     4501  1
;  4518     4502  2 BEGIN
;  4519     4503  2 MAP
;  4520     4504  2         FILE_ID:        REF BBLOCK,     ! Pointer to file ID
;  4521     4505  2         HEADER:         REF BBLOCK;     ! Pointer to file header
;  4522     4506  2 LOCAL
;  4523     4507  2         FILE_NUMBER,                    ! File number from file ID
;  4524     4508  2         RVN;                            ! RVN from file ID
;  4525     4509  2
;  4526     4510  2
;  4527     4511  2 ! Completely reinitialize the header if it is invalid.  Otherwise, use the
;  4528     4512  2 ! old copy to preserve the file sequence numbering.
;  4529     4513  2 !
;  4530     4514  2 IF VERIFY_HEADER(.HEADER, .FILE_ID) EQL 0
;  4531     4515  2 THEN
;  4532     4516  3     BEGIN
;  4533     4517  3     CH$FILL(0, 512, .HEADER);
;  4534     4518  3     HEADER[FH2$B_STRUCVER] = 1;
;  4535     4519  3     HEADER[FH2$B_STRUCLEV] = .STRUCTURE_LEVEL;
;  4536     4520  3     IF .STRUCTURE_LEVEL EQL 2
;  4537     4521  3         THEN HEADER[FH2$W_FID_SEQ] = .CURRENT_TIME<16,16>
;  4538     4522  3         ELSE HEADER[FH1$W_FID_SEQ] = .CURRENT_TIME<16,16>;
;  4539     4523  2     END;
;  4540     4524  2
;  4541     4525  2
;  4542     4526  2 ! Initialize the header as a valid deleted header.
;  4543     4527  2 !
;  4544     4528  2 IF .STRUCTURE_LEVEL EQL 2
;  4545     4529  2 THEN
;  4546     4530  3     BEGIN
```

```
;  4547          4531  3      HEADER[FH2$B_IDOFFSET] = FH2$C_LENGTH / 2;
;  4548          4532  3      HEADER[FH2$B_MPOFFSET] = (FH2$C_LENGTH + FI2$C_LENGTH) / 2;
;  4549          4533  3      HEADER[FH2$B_ACOFFSET] = $BYTEOFFSET (FH2$W_CHECKSUM) / 2;
;  4550          4534  3      HEADER[FH2$B_RSOFFSET] = $BYTEOFFSET (FH2$W_CHECKSUM) / 2;
;  4551          4535  3      HEADER[FH2$W_FID_SEQ] = .HEADER[FH2$W_FID_SEQ] + 1;
;  4552          4536  3      HEADER[FH2$W_FID_NUM] = 0;
;  4553          4537  3      HEADER[FH2$W_FID_RVN] = 0;
;  4554          4538  3      HEADER[FH2$W_CHECKSUM] = 0;
;  4555          4539  3      END
;  4556          4540  2  ELSE
;  4557          4541  3      BEGIN
;  4558          4542  3      HEADER[FH1$B_IDOFFSET] = FH1$C_LENGTH / 2;
;  4559          4543  3      HEADER[FH1$B_MPOFFSET] = (FH1$C_LENGTH + FI1$C_LENGTH) / 2;
;  4560          4544  3      HEADER[FH1$W_FID_SEQ] = .HEADER[FH1$W_FID_SEQ] + 1;
;  4561          4545  3      HEADER[FH1$W_FID_NUM] = 0;
;  4562          4546  3      HEADER[FH1$W_CHECKSUM] = 0;
;  4563          4547  3      END;
;  4564          4548  2
;  4565          4549  2
;  4566          4550  2  ! Write the new header, and clear the corresponding index file bitmap bit if
;  4567          4551  2  ! the write is successful.
;  4568          4552  2  !
;  4569          4553  2  IF WRITE_HEADER(.FILE_ID, .HEADER)
;  4570          4554  2  THEN
;  4571          4555  3      BEGIN
;  4572          4556  3      FILE_NUMBER = .FILE_ID[FID$W_NUM];
;  4573          4557  3      FILE_NUMBER<16,8> = .FILE_ID[FID$B_NMX];
;  4574          4558  3      RVN = .FILE_ID[FID$B_RVN];
;  4575          4559  3      BITVECTOR[.IMAP[.RVN-1], .FILE_NUMBER-1] = FALSE;
;  4576          4560  3      END;
;  4577          4561  1  END;


                                                01FC 00000 DELETE_HEADER:
                                                           .WORD   Save R2,R3,R4,R5,R6,R7,R8                    ;  4474
                                    58 00000000'  EF  9E 00002     MOVAB   STRUCTURE_LEVEL, R8
                                    57       04   AC  D0 00009     MOVL    FILE_ID, R7                          ;  4514
                                    56       08   AC  D0 0000D     MOVL    HEADER, R6
                                    7E            56  7D 00011     MOVQ    R6, -(SP)
                                F885  CF          02  FB 00014     CALLS   #2, VERIFY_HEADER
                                                  50  D5 00019     TSTL    R0
                                                  21  12 0001B     BNEQ    2$
                0200  8F          00        6E    00  2C 0001D     MOVC5   #0, (SP), #0, #512, (R6)             ;  4517
                                                  66  00024
                                  06  A6          01  90 00025     MOVB    #1, 6(R6)                            ;  4518
                                  07  A6          68  90 00029     MOVB    STRUCTURE_LEVEL, 7(R6)               ;  4519
                                      02          68  D1 0002D     CMPL    STRUCTURE_LEVEL, #2                  ;  4520
                                                  07  12 00030     BNEQ    1$
                                  0A  A6    12    A8  B0 00032     MOVW    CURRENT_TIME+2, 10(R6)               ;  4521
                                                  05  11 00037     BRB     2$
                                  04  A6    12    A8  B0 00039 1$:  MOVW    CURRENT_TIME+2, 4(R6)                ;  4522
                                      02          68  D1 0003E 2$:  CMPL    STRUCTURE_LEVEL, #2                  ;  4528
                                                  12  12 00041     BNEQ    3$
                                    66 FFFF6428   8F  D0 00043     MOVL    #-39896, (R6)                        ;  4531
```

```
                                    0A    A6   B6 0004A          INCW    10(R6)                                  ; 4535
                                    08    A6   B4 0004D          CLRW    8(R6)                                   ; 4536
                                    0C    A6   B4 00050          CLRW    12(R6)                                  ; 4537
                                          08   11 00053          BRB     4$                                      ; 4538
                                    04    A6   B6 00055 3$:      INCW    4(R6)                                   ; 4544
                              66    2E17  8F   3C 00058          MOVZWL  #11799, (R6)                            ; 4542
                                    01FE  C6   B4 0005D 4$:      CLRW    510(R6)                                 ; 4546
                                          56   DD 00061          PUSHL   R6                                      ; 4553
                                          57   DD 00063          PUSHL   R7
                        FF06  CF          02   FB 00065          CALLS   #2, WRITE_HEADER
                              19          50   E9 0006A          BLBC    R0, 5$
                              50          67   3C 0006D          MOVZWL  (R7), FILE_NUMBER                       ; 4556
        50      08            10    05    A7   F0 00070          INSV    5(R7), #16, #8, FILE_NUMBER             ; 4557
                              51    04    A7   9A 00076          MOVZBL  4(R7), RVN                              ; 4558
                              51    5C B841 DE   0007A          MOVAL   @IMAP[RVN], R1                          ; 4559
                                          50   D7 0007F          DECL    R0
                        00    FC    B1          50   E5 00081    BBCC    R0, @-4(R1), 5$
                                          04   00086 5$:      RET                                             ; 4561

; Routine Size:  135 bytes,    Routine Base:  CODE + 375C
```

```
;  4579     4562  1 ROUTINE CLEAR_EXT_FID(FILE_ID,HEADER): NOVALUE=
;  4580     4563  1
;  4581     4564  1 !++
;  4582     4565  1 !
;  4583     4566  1 ! FUNCTIONAL DESCRIPTION:
;  4584     4567  1 !       This routine clears the extension linkage in the specified header.
;  4585     4568  1 !
;  4586     4569  1 ! INPUT PARAMETERS:
;  4587     4570  1 !       FILE_ID         - File ID of the header.
;  4588     4571  1 !       HEADER          - Pointer to file header.
;  4589     4572  1 !
;  4590     4573  1 ! IMPLICIT INPUTS:
;  4591     4574  1 !       NONE
;  4592     4575  1 !
;  4593     4576  1 ! OUTPUT PARAMETERS:
;  4594     4577  1 !       NONE
;  4595     4578  1 !
;  4596     4579  1 ! IMPLICIT OUTPUTS:
;  4597     4580  1 !       NONE
;  4598     4581  1 !
;  4599     4582  1 ! ROUTINE VALUE:
;  4600     4583  1 !       NONE
;  4601     4584  1 !
;  4602     4585  1 ! SIDE EFFECTS:
;  4603     4586  1 !       File header written.
;  4604     4587  1 !
;  4605     4588  1 !--
;  4606     4589  1
;  4607     4590  2 BEGIN
;  4608     4591  2 MAP
;  4609     4592  2         FILE_ID:        REF BBLOCK,     ! Pointer to file ID
;  4610     4593  2         HEADER:         REF BBLOCK;     ! Pointer to file header
;  4611     4594  2 LOCAL
;  4612     4595  2         MAP_AREA:       REF BBLOCK;     ! Pointer to map area
;  4613     4596  2
;  4614     4597  2
;  4615     4598  2 ! Clear the extension linkage.
;  4616     4599  2 !
;  4617     4600  2 IF .STRUCTURE_LEVEL EQL 2
;  4618     4601  2 THEN
;  4619     4602  3     BEGIN
;  4620     4603  3     HEADER[FH2$W_EX_FIDNUM] = 0;
;  4621     4604  3     HEADER[FH2$W_EX_FIDSEQ] = 0;
;  4622     4605  3     HEADER[FH2$W_EX_FIDRVN] = 0;
;  4623     4606  3     END
;  4624     4607  2 ELSE
;  4625     4608  3     BEGIN
;  4626     4609  3     MAP_AREA = .HEADER + .HEADER[FH1$B_MPOFFSET]*2;
;  4627     4610  3     MAP_AREA[FM1$W_EX_FILNUM] = 0;
;  4628     4611  3     MAP_AREA[FM1$W_EX_FILSEQ] = 0;
;  4629     4612  3     END;
;  4630     4613  2
;  4631     4614  2
;  4632     4615  2 ! Rewrite the header.
;  4633     4616  2 !
;  4634     4617  2 WRITE_HEADER(.FILE_ID, .HEADER);
;  4635     4618  1 END;
```

```
                                 0000 00000 CLEAR_EXT_FID:
                                                .WORD   Save nothing                    ; 4562
                    51      08   AC  D0 00002           MOVL    HEADER, R1             ; 4603
                02 00000000'    EF  D1 00006           CMPL    STRUCTURE_LEVEL, #2    ; 4600
                                08  12 000CD           BNEQ    1$                     ;
                            0E  A1  D4 0000F           CLRL    14(R1)                 ; 4603
                            12  A1  B4 00012           CLRW    18(R1)                 ; 4605
                                0B  11 00015           BRB     2$                     ; 4600
                    50      01  A1  9A 00017 1$:       MOVZBL  1(R1), R0              ; 4609
                    50          6140 3E 0001B          MOVAW   (R1)[R0], MAP_AREA     ;
                            02  A0  D4 0001F           CLRL    2(MAP_AREA)            ; 4610
                                51  DD 00022 2$:       PUSHL   R1                     ; 4617
                            04  AC  DD 00024           PUSHL   FILE_ID                ;
                        FEBD    CF  02  FB 00027       CALLS   #2, @WRITE_HEADER      ;
                                    04  0002C         RET                            ; 4618

; Routine Size:  45 bytes,    Routine Base:  CODE + 37E3
```

```
;  4637     4619  1 ROUTINE READ_CHECK(FILE_ID,HEADER): NOVALUE=
;  4638     4620  1
;  4639     4621  1 !++
;  4640     4622  1 !
;  4641     4623  1 ! FUNCTIONAL DESCRIPTION:
;  4642     4624  1 !       This routine read-checks the indicated file.
;  4643     4625  1 !
;  4644     4626  1 ! INPUT PARAMETERS:
;  4645     4627  1 !       FILE_ID         - File ID of the file.
;  4646     4628  1 !       HEADER          - Header for the file.
;  4647     4629  1 !
;  4648     4630  1 ! IMPLICIT INPUTS:
;  4649     4631  1 !       TOTAL_SIZE      - Size of the file, determined from map area.
;  4650     4632  1 !
;  4651     4633  1 ! OUTPUT PARAMETERS:
;  4652     4634  1 !       NONE
;  4653     4635  1 !
;  4654     4636  1 ! IMPLICIT OUTPUTS:
;  4655     4637  1 !       NONE
;  4656     4638  1 !
;  4657     4639  1 ! ROUTINE VALUE:
;  4658     4640  1 !       NONE
;  4659     4641  1 !
;  4660     4642  1 ! SIDE EFFECTS:
;  4661     4643  1 !       All allocated blocks of the file read and errors reported.
;  4662     4644  1 !
;  4663     4645  1 !--
;  4664     4646  1
;  4665     4647  2 BEGIN
;  4666     4648  2 MAP
;  4667     4649  2         FILE_ID:        REF BBLOCK,     ! Pointer to file ID
;  4668     4650  2         HEADER:         REF BBLOCK;     ! Pointer to file header
;  4669     4651  2 LOCAL
;  4670     4652  2         BUFFER_SIZE,                    ! Size of file buffer
;  4671     4653  2         BUFFER_ADDRESS,                 ! Address of file buffer
;  4672     4654  2         VBN,                            ! Current VBN
;  4673     4655  2         STATUS;                         ! Status variable
;  4674     4656  2
;  4675     4657  2
;  4676     4658  2 ! If file size is zero, there is nothing to do.
;  4677     4659  2 !
;  4678     4660  2 IF .TOTAL_SIZE EQL 0 THEN RETURN;
;  4679     4661  2
;  4680     4662  2
;  4681     4663  2 ! Access the file being read-checked.  Call ACCESS_INDEX_2 to deaccess the
;  4682     4664  2 ! second channel in case it is being used.  If the access fails, report it
;  4683     4665  2 ! and quit.
;  4684     4666  2 !
;  4685     4667  2 ACCESS_INDEX_2(0);
;  4686     4668  2 CH$FILL(0, FIB$C_LENGTH, FIB);
;  4687     4669  2 FIB[FIB$L_ACCTL] = FIB$M_NORECORD;
;  4688     4670  2 FIB[FIB$W_FID_NUM] = .FILE_ID[FID$W_NUM];
;  4689     4671  2 FIB[FIB$W_FID_SEQ] = .FILE_ID[FID$W_SEQ];
;  4690     4672  2 FIB[FIB$W_FID_RVN] = .FILE_ID[FID$W_RVN];
;P 4691     4673  2 STATUS = $QIOW(
;P 4692     4674  2     FUNC=IO$_ACCESS OR IO$M_ACCESS,
;P 4693     4675  2     CHAN=.CHANNEL_2,
```

```
;  4694        P 4676  2      IOSB=IOSB,
;  4695          4677  2      P1=FIB_DESC);
;  4696          4678  2  IF .STATUS THEN STATUS = .IOSB[0];
;  4697          4679  2  IF NOT .STATUS
;  4698          4680  2  THEN
;  4699          4681  2      BEGIN
;  4700          4682  2      HEADER_ERROR(VERIFY$_OPENFILE, .FILE_ID, .HEADER, .STATUS);
;  4701          4683  2      RETURN;
;  4702          4684  2      END;
;  4703          4685  2
;  4704          4686  2
;  4705          4687  2  ! Allocate the buffer.  Use the size of the file, thresholded by
;  4706          4688  2  ! FILE_BUF_COUNT.
;  4707          4689  2  !
;  4708          4690  2  BUFFER_SIZE = MINU(FILE_BUF_COUNT, .TOTAL_SIZE);
;  4709          4691  2  STATUS = LIB$GET_VM(%REF(.BUFFER_SIZE * 512), BUFFER_ADDRESS);
;  4710          4692  2  IF NOT .STATUS THEN SIGNAL(VERIFY$_ALLOCMEM, 0, .STATUS);
;  4711          4693  2
;  4712          4694  2
;  4713          4695  2  ! Loop over all blocks of the file.
;  4714          4696  2  !
;  4715          4697  2  VBN = 1;
;  4716          4698  2  WHILE .VBN LEQU .TOTAL_SIZE DO
;  4717          4699  2      BEGIN
;  4718          4700  2      LOCAL
;  4719          4701  2          THIS_BLOCKS;    ! Count of blocks to read on current iteration
;  4720          4702  2
;  4721          4703  2
;  4722          4704  2      ! Compute how many blocks to read on this iteration and read them.
;  4723          4705  2      !
;  4724          4706  2      THIS_BLOCKS = MINU(.BUFFER_SIZE, .TOTAL_SIZE + 1 - .VBN);
;  4725        P 4707  2      STATUS = $QIOW(
;  4726        P 4708  2          FUNC=IO$_READVBLK,
;  4727        P 4709  2          CHAN=.CHANNEL_2,
;  4728        P 4710  2          IOSB=IOSB,
;  4729        P 4711  2          P1=.BUFFER_ADDRESS,
;  4730        P 4712  2          P2=.THIS_BLOCKS * 512,
;  4731          4713  2          P3=.VBN);
;  4732          4714  2      IF .STATUS THEN STATUS = .IOSB[0];
;  4733          4715  2
;  4734          4716  2
;  4735          4717  2      ! If an error occurred, read each block individually, reporting errors
;  4736          4718  2      ! as we go.
;  4737          4719  2      !
;  4738          4720  2      IF NOT .STATUS
;  4739          4721  2      THEN
;  4740          4722  2          BEGIN
;  4741          4723  3          INCR XVBN FROM 0 TO .THIS_BLOCKS-1 DO
;  4742          4724  3              BEGIN
;  4743        P 4725  3              STATUS = $QIOW(
;  4744        P 4726  3                  FUNC=IO$_READVBLK,
;  4745        P 4727  3                  CHAN=.CHANNEL_2,
;  4746        P 4728  3                  IOSB=IOSB,
;  4747        P 4729  3                  P1=.BUFFER_ADDRESS,
;  4748        P 4730  3                  P2=512,
;  4749          4731  3                  P3=.VBN + .XVBN);
;  4750          4732  3              IF .STATUS THEN STATUS = .IOSB[0];
```

```
;  4751          4733  5              IF NOT .STATUS
;  4752          4734  5              THEN
;  4753          4735  5                  HEADER_ERROR(
;  4754          4736  5                      VERIFY$_READFILE,
;  4755          4737  5                      .FILE_ID, .HEADER,
;  4756          4738  5                      .VBN + .XVBN, .STATUS);
;  4757          4739  4              END;
;  4758          4740  3          END;
;  4759          4741  2      VBN = .VBN + .THIS_BLOCKS;
;  4760          4742  2      END;
;  4761          4743  2
;  4762          4744  2
;  4763          4745  2  ! Free the buffer.
;  4764          4746  2  !
;  4765          4747  2  STATUS = LIB$FREE_VM(%REF(.BUFFER_SIZE * 512), BUFFER_ADDRESS);
;  4766          4748  2  IF NOT .STATUS THEN SIGNAL(VERIFY$_FREEMEM, 0, .STATUS);
;  4767          4749  2
;  4768          4750  2
;  4769          4751  2  ! Deaccess the file being read-checked.
;  4770          4752  2  !
;  4771        P 4753  2  $QIOW(
;  4772        P 4754  2      FUNC=IO$_DEACCESS,
;  4773          4755  2      CHAN=.CHANNEL_2);
;  4774          4756  1  END;


                                                OFFC 00000 READ_CHECK:
                                                           .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11          ; 4619
                                    5B 00000000G   00   9E 00002          MOVAB   LIB$SIGNAL, R11
                                    5A 00000000G   00   9E 00009          MOVAB   SYS$QIOW, R10
                                    59 00000000'   EF   9E 00010          MOVAB   IOSB, R9
                                    5E             08   C2 00017          SUBL2   #8, SP
                                       00000000'   EF   D5 0001A          TSTL    TOTAL_SIZE                    ; 4660
                                                   01   12 00020          BNEQ    1$
                                                        04 00022          RET
                                                   7E   D4 00023 1$:      CLRL    -(SP)                         ; 4667
                                          FCA7     CF   01   FB 00025          CALLS   #1, ACCESS_INDEX_2
0040    8F              00                 6E        00   2C 0002A          MOVC5   #0, (SP), #0, #64, FIB        ; 4668
                                                 FE44   C9      00031
                                    FE44     C9 00200000   8F   D0 00034          MOVL    #2097152, FIB         ; 4669
                                    56             04    AC   D0 0003D          MOVL    FILE_ID, R6           ; 4670
                                    FE48     C9          66   D0 00041          MOVL    (R6), FIB+4
                                    FE4C     C9          A6   B0 00046          MOVW    4(R6), FIB+8          ; 4672
                                                         7E   7C 0004C          CLRQ    -(SP)                 ; 4677
                                                         7E   7C 0004E          CLRQ    -(SP)
                                                         7E   D4 00050          CLRL    -(SP)
                                                   C7CA  CF   9F 00052          PUSHAB  FIB_DESC
                                                         7E   7C 00056          CLRQ    -(SP)
                                                         59   DD 00058          PUSHL   R9
                                                   7E       72   8F   9A 0005A          MOVZBL  #114, -(SP)
                                                      00000000'   EF   DD 0005E          PUSHL   CHANNEL_2
                                                         7E   D4 00064          CLRL    -(SP)
                                                   6A         0C   FB 00066          CALLS   #12, SYS$QIOW
                                                   55         50   D0 00069          MOVL    R0, STATUS
```

```
                        06              55  E9 0006C            BLBC    STATUS, 2$                      ;4678
                        55              69  3C 0006F            MOVZWL  IOSB, STATUS                    ;
                        13              55  E8 00072            BLBS    STATUS, 3$                      ;4679
                                        55  DD 00075 2$:        PUSHL   STATUS                          ;4682
                                08      AC  DD 00077            PUSHL   HEADER                          ;
                                        56  DD 0007A            PUSHL   R6                              ;
                            00000000G   8F  DD 0007C            PUSHL   #VERIFY$_OPENFILE               ;
                    0000V   CF          04  FB 00082            CALLS   #4, HEADER_ERROR                ;
                                            04 00087            RET                                     ;4681
                        50  00000000'   EF  D0 00088 3$:        MOVL    TOTAL_SIZE, R0                  ;4690
                00000040    8F          50  D1 0008F            CMPL    R0, #64                         ;
                                            04  1B 00096        BLEQU   4$                              ;
                        50          40  8F  9A 00098            MOVZBL  #64, R0                         ;
                        52              50  D0 0009C 4$:        MOVL    R0, BUFFER_SIZE                 ;
                                    04  AE  9F 0009F            PUSHAB  BUFFER_ADDRESS                  ;4691
                58      52              09  78 000A2            ASHL    #9, BUFFER_SIZE, R8             ;
                    04  AE              58  D0 000A6            MOVL    R8, 4(SP)                       ;
                                    04  AE  9F 000AA            PUSHAB  4(SP)                           ;
                00000000G   00          02  FB 000AD            CALLS   #2, LIB$GET_VM                  ;
                        55              50  D0 000B4            MOVL    R0, STATUS                      ;
                        0D              55  E8 000B7            BLBS    STATUS, 5$                      ;4692
                                        55  DD 000BA            PUSHL   STATUS                          ;
                                        7E  D4 000BC            CLRL    -(SP)                           ;
                            00000000G   8F  DD 000BE            PUSHL   #VERIFY$_ALLOCMEM               ;
                        6B              03  FB 000C4            CALLS   #3, LIB$SIGNAL                  ;
                        53              01  D0 000C7 5$:        MOVL    #1, VBN                         ;4697
                00000000'   EF          53  D1 000CA 6$:        CMPL    VBN, TOTAL_SIZE                 ;4698
                                            03  1B 000D1        BLEQU   7$                              ;
                                        0092 31 000D3           BRW     14$                             ;
            51  00000000'   EF          53  C3 000D6 7$:        SUBL3   VBN, TOTAL_SIZE, R1             ;4706
                                            51  D6 000DE        INCL    R1                              ;
                        50              52  D0 000E0            MOVL    BUFFER_SIZE, R0                 ;
                        51              50  D1 000E3            CMPL    R0, R1                          ;
                                            03  1B 000E6        BLEQU   8$                              ;
                        50              51  D0 000E8            MOVL    R1, R0                          ;
                        57              50  D0 000EB 8$:        MOVL    R0, THIS_BLOCKS                 ;
                                        7E  7C 000EE            CLRQ    -(SP)                           ;4713
                                        7E  D4 000F0            CLRL    -(SP)                           ;
                                        53  DD 000F2            PUSHL   VBN                             ;
                7E      57              09  78 000F4            ASHL    #9, THIS_BLOCKS, -(SP)          ;
                            18      AE  DD 000F8            PUSHL   BUFFER_ADDRESS                  ;
                                        7E  7C 000FB            CLRQ    -(SP)                           ;
                                        59  DD 000FD            PUSHL   R9                              ;
                                        31  DD 000FF            PUSHL   #49                             ;
                            00000000'   EF  DD 00101            PUSHL   CHANNEL_2                       ;
                                        7E  D4 00107            CLRL    -(SP)                           ;
                        6A              0C  FB 00109            CALLS   #12, SYS$QIOW                   ;
                        55              50  D0 0010C            MOVL    R0, STATUS                      ;
                        06              55  E9 0010F            BLBC    STATUS, 9$                      ;4714
                        55              69  3C 00112            MOVZWL  IOSB, STATUS                    ;
                        4A              55  E8 00115            BLBS    STATUS, 13$                     ;4720
                        54              01  CE 00118 9$:        MNEGL   #1, XVBN                        ;4723
                                            41  11 0011B        BRB     12$                             ;
                                        7E  7C 0011D 10$:       CLRQ    -(SP)                           ;4731
                                        7E  D4 0011F            CLRL    -(SP)                           ;
                                    6443 9F 00121               PUSHAB  (XVBN)[VBN]                     ;
                        7E      0200    8F  3C 00124            MOVZWL  #512, -(SP)                     ;
```

```
                                 18   AE   DD 00129          PUSHL   BUFFER_ADDRESS                                  ;
                                      7E   7C 0012C          CLRQ    -(SP)                                          ;
                                      59   DD 0012E          PUSHL   R9                                             ;
                                      31   DD 00130          PUSHL   #49                                            ;
                           00000000'  EF   DD 00132          PUSHL   CHANNEL_2                                      ;
                                      7E   D4 00138          CLRL    -(SP)                                          ;
                              6A      0C   FB 0013A          CALLS   #12, SYS$QIOW                                  ;
                              55      50   D0 0013D          MOVL    R0, STATUS                                     ;
                              06      55   E9 00140          BLBC    STATUS, 11$                                    ; 4732
                              55      69   3C 00143          MOVZWL  IOSB, STATUS                                   ;
                              15      55   E8 00146          BLBS    STATUS, 12$                                    ; 4733
                                      55   DD 00149 11$:      PUSHL   STATUS                                         ; 4738
                                 6443 9F   00 0014B          PUSHAB  (XVBN)[VBN]                                    ;
                                 08   AC   DD 0014E          PUSHL   HEADER                                         ; 4737
                                      56   DD 00151          PUSHL   R6                                             ;
                           00000000G  8F   DD 00153          PUSHL   #VERIFY$_READFILE                              ; 4735
                     0000V CF         05   FB 00159          CALLS   #5, HEADER_ERROR                               ;
                  BB      54          57   F2 0015E 12$:      AOBLSS  THIS_BLOCKS, XVBN, 10$                         ; 4723
                              53      57   C0 00162 13$:      ADDL2   THIS_BLOCKS, VBN                               ; 4741
                                 FF62 31   00 00165          BRW     6$                                             ; 4698
                                 04   AE   9F 00168 14$:      PUSHAB  BUFFER_ADDRESS                                 ; 4747
                           04    AE        58   D0 0016B          MOVL    R8, 4(SP)                                      ;
                                 04   AE   9F 0016F          PUSHAB  4(SP)                                          ;
                     00000000G     00        02   FB 00172          CALLS   #2, LIB$FREE_VM                                ;
                                      55   50   D0 00179          MOVL    R0, STATUS                                     ;
                                      0D   55   E8 0017C          BLBS    STATUS, 15$                                    ; 4748
                                      55   DD 0017F          PUSHL   STATUS                                         ;
                                      7E   D4 00181          CLRL    -(SP)                                          ;
                           00000000G  8F   DD 00183          PUSHL   #VERIFY$_FREEMEM                               ;
                              6B      03   FB 00189          CALLS   #3, LIB$SIGNAL                                 ;
                                      7E   7C 0018C 15$:      CLRQ    -(SP)                                          ; 4755
                                      7E   7C 0018E          CLRQ    -(SP)                                          ;
                                      7E   7C 00190          CLRQ    -(SP)                                          ;
                                      7E   7C 00192          CLRQ    -(SP)                                          ;
                              7E      34   7D 00194          MOVQ    #52, -(SP)                                     ;
                           00000000'  EF   DD 00197          PUSHL   CHANNEL_2                                      ;
                                      7E   D4 0019D          CLRL    -(SP)                                          ;
                              6A      0C   FB 0019F          CALLS   #12, SYS$QIOW                                  ;
                                           04   00 001A2          RET                                                    ; 4756

; Routine Size:  419 bytes,    Routine Base:  CODE + 3810
```

```
;  4776     4757  1 ROUTINE FILE_ERROR(MESSAGE,FAB,EXTRA1,EXTRA2): NOVALUE=
;  4777     4758  1
;  4778     4759  1 !++
;  4779     4760  1 !
;  4780     4761  1 ! FUNCTIONAL DESCRIPTION:
;  4781     4762  1 !       This routine signals a file-related message.
;  4782     4763  1 !
;  4783     4764  1 ! INPUT PARAMETERS:
;  4784     4765  1 !       MESSAGE         - Message code for first message
;  4785     4766  1 !       FAB             - Pointer to FAB, from which file name
;  4786     4767  1 !                         will be obtained
;  4787     4768  1 !       Up to two additional input parameters are additional messages --
;  4788     4769  1 !       except if the message is one of the special cases, they are
;  4789     4770  1 !       additional FAO arguments.
;  4790     4771  1 !
;  4791     4772  1 ! IMPLICIT INPUTS:
;  4792     4773  1 !       NONE
;  4793     4774  1 !
;  479-     4775  1 ! OUTPUT PARAMETERS:
;  4795     4776  1 !       NONE
;  4796     4777  1 !
;  4797     4778  1 ! IMPLICIT OUTPUTS:
;  4798     4779  1 !       NONE
;  4799     4780  1 !
;  4800     4781  1 ! ROUTINE VALUE:
;  4801     4782  1 !       NONE
;  4802     4783  1 !
;  4803     4784  1 ! SIDE EFFECTS:
;  4804     4785  1 !       The messages are signalled.
;  4805     4786  1 !
;  4806     4787  1 !--
;  4807     4788  1
;  4808     4789  2 BEGIN
;  4809     4790  2 MAP
;  4810     4791  2         FAB:            REF BBLOCK;     ! Pointer to FAB
;  4811     4792  2 LOCAL
;  4812     4793  2         NAM:            REF BBLOCK,     ! Pointer to NAM block
;  4813     4794  2         DESC:           BBLOCK[8],      ! Descriptor for signal
;  4814     4795  2         PARAM:          VECTOR[6];      ! Signal parameter list
;  4815     4796  2 BUILTIN
;  4816     4797  2         ACTUALCOUNT;
;  4817     4798  2
;  4818     4799  2
;  4819     4800  2 NAM = .FAB[FAB$L_NAM];
;  4820     4801  2 DESC[DSC$B_DTYPE] = 0;
;  4821     4802  2 DESC[DSC$B_CLASS] = 0;
;  4822     4803  2 IF .NAM[NAM$B_RSL] NEQ 0
;  4823     4804  2 THEN
;  4824     4805  3     BEGIN
;  4825     4806  3     DESC[DSC$W_LENGTH] = .NAM[NAM$B_RSL];
;  4826     4807  3     DESC[DSC$A_POINTER] = .NAM[NAM$[_RSA];
;  4827     4808  3     END
;  4828     4809  2 ELSE IF .NAM[NAM$B_ESL] NEQ 0
;  4829     4810  2 THEN
;  4830     4811  3     BEGIN
;  4831     4812  3     DESC[DSC$W_LENGTH] = .NAM[NAM$B_ESL];
;  4832     4813  3     DESC[DSC$A_POINTER] = .NAM[NAM$[_ESA];
```

```
;  4833     4814  3      END
;  4834     4815  2  ELSE
;  4835     4816  2      BEGIN
;  4836     4817  2      DESC[DSC$W_LENGTH] = .FAB[FAB$B_FNS];
;  4837     4818  2      DESC[DSC$A_POINTER] = .FAB[FAB$L_FNA];
;  4838     4819  2      END;
;  4839     4820  2
;  4840     4821  2
;  4841     4822  2  PARAM[0] = 3;
;  4842     4823  2  PARAM[1] = .MESSAGE;
;  4843     4824  2  PARAM[2] = 1;
;  4844     4825  2  PARAM[3] = DESC;
;  4845     4826  2  IF ACTUALCOUNT() GEQ 3
;  4846     4827  2  THEN
;  4847     4828  3      BEGIN
;  4848     4829  3      PARAM[0] = .PARAM[0] + 1;
;  4849     4830  3      PARAM[4] = .EXTRA1;
;  4850     4831  2      END;
;  4851     4832  2  IF ACTUALCOUNT() GEQ 4
;  4852     4833  2  THEN
;  4853     4834  3      BEGIN
;  4854     4835  3      PARAM[0] = .PARAM[0] + 1;
;  4855     4836  3      PARAM[5] = .EXTRA2;
;  4856     4837  2      END;
;  4857     4838  2
;  4858     4839  2
;  4859     4840  2  CALLG(PARAM, LIB$SIGNAL);
;  4860     4841  1  END;


                        0000 00000 FILE_ERROR:
                                   .WORD   Save nothing                    ;  4757
                5E      20  C2 00002          SUBL2   #32, SP
                51  08  AC  D0 00005          MOVL    FAB, R1                ;  4800
                50  28  A1  D0 00009          MOVL    40(R1), NAM
                    1A  AE  B4 0000D          CLRW    DESC+2                 ;  4801
                    03  A0  95 00010          TSTB    3(NAM)                 ;  4803
                        0C  13 00013          BEQL    1$
        18  AE      03  A0  9B 00015          MOVZBW  3(NAM), DESC           ;  4806
        1C  AE      04  A0  D0 0001A          MOVL    4(NAM), DESC+4         ;  4807
                        1B  11 0001F          BRB     3$                     ;  4803
                    0B  A0  95 00021  1$:      TSTB    11(NAM)                ;  4809
                        0C  13 00024          BEQL    2$
        18  AE      0B  A0  9B 00026          MOVZBW  11(NAM), DESC          ;  4812
        1C  AE      0C  A0  D0 0002B          MOVL    12(NAM), DESC+4        ;  4813
                        0A  11 00030          BRB     3$                     ;  4809
        18  AE      34  A1  9B 00032  2$:      MOVZBW  52(R1), DESC           ;  4817
        1C  AE      2C  A1  D0 00037          MOVL    44(R1), DESC+4         ;  4818
            6E          03  D0 0003C  3$:      MOVL    #3, PARAM              ;  4822
        04  AE      04  AC  D0 0003F          MOVL    MESSAGE, PARAM+4       ;  4823
        08  AE          01  D0 00044          MOVL    #1, PARAM+8            ;  4824
        0C  AE      18  AE  9E 00048          MOVAB   DESC, PARAM+12         ;  4825
            03          6C  91 0004D          CMPB    (AP), #3               ;  4826
                        07  1F 00050          BLSSU   4$                     ;
```

```
                                 6E  D6 00052              INCL    PARAM                                   ; 4829
                          10     AE      0C  AC  D0 00054        MOVL    EXTRA1, PARAM+16                        ; 4830
                          04                 6C  91 00059 4$:     CMPB    (AP), #4                                ; 4832
                                             07  1F 0005C        BLSSU   5$                                      ;
                                             6E  D6 0005E        INCL    PARAM                                   ; 4835
                          14     AE      10  AC  D0 00060        MOVL    EXTRA2, PARAM+20                        ; 4836
                    00000000G    00          6E  FA 00065 5$:     CALLG   PARAM, LIB$SIGNAL                       ; 4840
                                             04 0006C            RET                                             ; 4841

; Routine Size:  109 bytes,    Routine Base:  CODE + 3983
```

```
;  4862     4842  1 ROUTINE HEADER_ERROR(MESSAGE,FILE_ID,HEADER,EXTRA1,EXTRA2,EXTRA3,EXTRA4,EXTRA5): NOVALUE=
;  4863     4843  1
;  4864     4844  1 !++
;  4865     4845  1 !
;  4866     4846  1 ! FUNCTIONAL DESCRIPTION:
;  4867     4847  1 !       This routine signals a header-related message.
;  4868     4848  1 !
;  4869     4849  1 ! INPUT PARAMETERS:
;  4870     4850  1 !       MESSAGE         - Message code for first message
;  4871     4851  1 !       FILE_ID         - Pointer to file ID
;  4872     4852  1 !       HEADER          - Pointer to header in memory or zero if not
;  4873     4853  1 !       EXTRA...        - Extra parameters, depending on message
;  4874     4854  1 !
;  4875     4855  1 ! IMPLICIT INPUTS:
;  4876     4856  1 !       NONE
;  4877     4857  1 !
;  4878     4858  1 ! OUTPUT PARAMETERS:
;  4879     4859  1 !       NONE
;  4880     4860  1 !
;  4881     4861  1 ! IMPLICIT OUTPUTS:
;  4882     4862  1 !       NONE
;  4883     4863  1 !
;  4884     4864  1 ! ROUTINE VALUE:
;  4885     4865  1 !       NONE
;  4886     4866  1 !
;  4887     4867  1 ! SIDE EFFECTS:
;  4888     4868  1 !       The message is signalled.
;  4889     4869  1 !
;  4890     4870  1 !--
;  4891     4871  1
;  4892     4872  2 BEGIN
;  4893     4873  2 MAP
;  4894     4874  2         FILE_ID:        REF BBLOCK;             ! Pointer to file ID
;  4895     4875  2 LOCAL
;  4896     4876  2         FILENAME:       VECTOR[FI2$S_FILENAME + FI2$S_FILENAMEXT + 1, BYTE],
;  4897     4877  2                                                 ! Buffer for ASCII filename
;  4898     4878  2         PARAM:          VECTOR[13],             ! Signal parameters
;  4899     4879  2         BUFFER:         BBLOCK[512],            ! Header buffer
;  4900     4880  2         HDR:            REF BBLOCK;             ! Pointer to good header
;  4901     4881  2
;  4902     4882  2
;  4903     4883  2 ! If the file sequence number is 65535, cover the error.  This is probably a
;  4904     4884  2 ! file header that we deleted earlier.
;  4905     4885  2 !
;  4906     4886  2 IF .FILE_ID[FID$W_SEQ] EQL 65535
;  4907     4887  2 THEN
;  4908     4888  2     RETURN;
;  4909     4889  2
;  4910     4890  2
;  4911     4891  2 ! Initialize the beginning of the signal vector.
;  4912     4892  2 !
;  4913     4893  2 PARAM[0] = 7;
;  4914     4894  2 PARAM[1] = .MESSAGE;
;  4915     4895  2 PARAM[2] = 5;
;  4916     4896  2 PARAM[3] = .FILE_ID[FID$W_NUM] + .FILE_ID[FID$B_NMX] ^ 16;
;  4917     4897  2 PARAM[4] = .FILE_ID[FID$W_SEQ];
;  4918     4898  2 PARAM[5] = .FILE_ID[FID$B_RVN];
```

```
;  4919     4899  2
;  4920     4900
;  4921     4901  ! Get the file name.  If no header was passed, read the header.
;  4922     4902  ! If reading the header fails, use a null string.
;  4923     4903  !
;  4924     4904  IF
;  4925     4905      BEGIN
;  4926     4906      HDR = .HEADER;
;  4927     4907      IF .HDR EQL 0
;  4928     4908      THEN
;  4929     4909  4       BEGIN
;  4930     4910  4       HDR = BUFFER;
;  4931     4911  4       READ_HEADER(.FILE_ID, .HDR)
;  4932     4912  4       END
;  4933     4913      ELSE
;  4934     4914  4       TRUE
;  4935     4915      END
;  4936     4916  THEN
;  4937     4917      BEGIN
;  4938     4918      LOCAL
;  4939     4919  1       IDENT_AREA:     REF BBLOCK;     ! Pointer to ident area
;  4940     4920  1
;  4941     4921  1   IDENT_AREA = .HDR + .HDR[FH2$B_IDOFFSET]*2;
;  4942     4922  1   IF .STRUCTURE_LEVEL EQL 2
;  4943     4923  1   THEN
;  4944     4924  1       BEGIN
;  4945     4925  1       CH$COPY(
;  4946     4926  1           FI2$S_FILENAME, IDENT_AREA[FI2$T_FILENAME],
;  4947     4927  1           %C' ',
;  4948     4928  1           FI2$S_FILENAME + FI2$S_FILENAMEXT + 1, FILENAME);
;  4949     4929  1
;  4950     4930  1       IF (.HDR[FH2$B_MPOFFSET] - .HDR[FH2$B_IDOFFSET]) * 2
;  4951     4931  1       GEQU $BYTEOFFSET(FI2$T_FILENAMEXT) + FI2$S_FILENAMEXT
;  4952     4932  1       THEN
;  4953     4933  1           CH$MOVE(
;  4954     4934  1               FI2$S_FILENAMEXT,
;  4955     4935  1               IDENT_AREA[FI2$T_FILENAMEXT],
;  4956     4936  1               FILENAME[FI2$S_FILENAME]);
;  4957     4937  1
;  4958     4938  1       PARAM[6] =
;  4959     4939  1           CH$FIND_CH(FI2$S_FILENAME + FI2$S_FILENAMEXT + 1, FILENAME, %C' ')
;  4960     4940  1           - FILENAME;
;  4961     4941  1       END
;  4962     4942  3   ELSE
;  4963     4943  1       BEGIN
;  4964     4944  1       PARAM[6] = MAKE_STRING(
;  4965     4945  1           IDENT_AREA[FI1$W_FILENAME] - $BYTEOFFSET(NMB$W_NAME),
;  4966     4946  1           FILENAME);
;  4967     4947  3       END;
;  4968     4948  3   PARAM[7] = FILENAME;
;  4969     4949  3   END
;  4970     4950  2 ELSE
;  4971     4951  3   BEGIN
;  4972     4952  3   PARAM[6] = 0;
;  4973     4953  3   PARAM[7] = .HDR;
;  4974     4954  2   END;
;  4975     4955  2
```

```
2
2 ! Handle a variety of special cases for the message code.
2 !
2 IF
2     .MESSAGE EQL VERIFY$_READHEADER OR
2     .MESSAGE EQL VERIFY$_WRITEHEADER OR
2     .MESSAGE EQL VERIFY$_OPENFILE OR
2     .MESSAGE EQL VERIFY$_ENTERLOST OR
2     .MESSAGE EQL VERIFY$_DELETE
2 THEN
3     BEGIN
3     PARAM[0] = 8;
3     PARAM[8] = .EXTRA1;
3     END
2 ELSE IF
2     .MESSAGE EQL VERIFY$_READFILE
2 THEN
3     BEGIN
3     PARAM[0] = 9;
3     PARAM[2] = 6;
3     PARAM[8] = .EXTRA1;
3     PARAM[9] = .EXTRA2;
3     END
2 ELSE IF
2     .MESSAGE EQL VERIFY$_MULTALLOC
2 THEN
3     BEGIN
3     PARAM[0] = 12;
3     PARAM[2] = 10;
3     PARAM[8] = .EXTRA1;
3     PARAM[9] = .EXTRA2;
3     PARAM[10] = .EXTRA3;
3     PARAM[11] = .EXTRA4;
3     PARAM[12] = .EXTRA5;
3     END;
2
2 ! Finally, signal the message.
2 !
2 CALLG(PARAM, LIB$SIGNAL);
1 END;
```

```
                                   03FC 00000 HEADER_ERROR:
                                                  .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9      ; 4842
            5E     FD74   CE  9E 00002            MOVAB   -652(SP), SP                      ;
            50       08   AC  D0 00007            MOVL    FILE_ID, R0                       ; 4886
     FFFF   8F       02   A0  B1 0000B            CMPW    2(R0), #65535                     ;
                          01  12 00011            BNEQ    1$                                ;
                              04 00013            RET                                       ;
     FF74   CD            07  D0 00014 1$:        MOVL    #7, PARAM                         ; 4893
            59       04   AC  D0 00019            MOVL    MESSAGE, R9                       ; 4894
     FF78   CD            59  D0 0001D            MOVL    R9, PARAM+4                       ;
     FF7C   CD            05  D0 00022            MOVL    #5, PARAM+8                       ; 4895
```

```
                         51      05   A0  9A 00027          MOVZBL  5(R0), R1                                   ; 4896
             51          51           10  78 0002B          ASHL    #16, R1, R1                                 ;
                         52           60  3C 0002F          MOVZWL  (R0), R2                                    ;
        80   AD          51           52  C1 00032          ADDL3   R2, R1, PARAM+12                            ;
                  84     AD      02   A0  3C 00037          MOVZWL  2(R0), PARAM+16                             ; 4897
                  88     AD      04   A0  9A 0003C          MOVZBL  4(R0), PARAM+20                             ; 4898
                         56      0C   AC  D0 00041          MOVL    HEADER, HDR                                 ; 4906
                                      0F  12 00045          BNEQ    2$                                          ; 4907
                         56           6E  9E 00047          MOVAB   BUFFER, HDR                                 ; 4910
                               0041   8F  BB 0004A          PUSHR   #^M<R0,R6>                                  ; 4911
                FBB3     CF           02  FB 0004E          CALLS   #2, READ_HEADER                             ;
                         62           50  E9 00053          BLBC    R0, 7$                                      ;
                         57           66  9A 00056 2$:      MOVZBL  (HDR), R7                                   ; 4921
                         58         6647  3E 00059          MOVAW   (HDR)[R7], IDENT_AREA                       ;
                         02 00000000' EF  D1 0005D          CMPL    STRUCTURE_LEVEL, #2                         ; 4922
                                      3A  12 00064          BNEQ    5$                                          ;
 0057   8F       20      68           14  2C 00066          MOVC5   #20, (IDENT_AREA), #32, #87, FILENAME       ; 4926
                                 A8   AD     0006D                                                              ;
                         56      01   A6  9A 0006F          MOVZBL  1(HDR), R6                                  ; 4930
                         56           57  C2 00073          SUBL2   R7, R6                                      ;
                         56           02  C4 00076          MULL2   #2, R6                                      ;
              00000078   8F           56  D1 00079          CMPL    R6, #120                                    ; 4931
                                      08  1F 00080          BLSSU   3$                                          ;
        BC   AD          36  A8 0042  8F  28 00082          MOVC3   #66, 54(IDENT_AREA), FILENAME+20            ; 4936
        A8   AD        0057  8F       20  3A 0008A 3$:      LOCC    #32, #87, FILENAME                          ; 4939
                                      02  12 00091          BNEQ    4$                                          ;
                                      51  D4 00093          CLRL    R1                                          ;
                         50      A8   AD  9E 00095 4$:      MOVAB   FILENAME, R0                                ; 4940
        8C   AD          51           50  C3 00099          SUBL3   R0, R1, PARAM+24                            ;
                                      11  11 0009E          BRB     6$                                          ; 4922
                                 A8   AD  9F 000A0 5$:      PUSHAB  FILENAME                                    ; 4944
                                 FA   A8  9F 000A3          PUSHAB  -6(IDENT_AREA)                              ; 4945
             00000000G   EF           02  FB 000A6          CALLS   #2, MAKE_STRING                             ;
                    8C   AD           50  D0 000AD          MOVL    R0, PARAM+24                                ;
                    90   AD      A8   AD  9E 000B1 6$:      MOVAB   FILENAME, PARAM+28                          ; 4948
                                      07  11 000B6          BRB     8$                                          ; 4904
                                 8C   AD  D4 000B8 7$:      CLRL    PARAM+24                                    ; 4952
                    90   AD           56  D0 000BB          MOVL    HDR, PARAM+28                               ; 4953
             00000000G   8F           59  D1 000BF 8$:      CMPL    R9, #VERIFY$_READHEADER                     ; 4960
                                      24  13 000C6          BEQL    9$                                          ;
             00000000G   8F           59  D1 000C8          CMPL    R9, #VERIFY$_WRITEHEADER                    ; 4961
                                      1B  13 000CF          BEQL    9$                                          ;
             00000000G   8F           59  D1 000D1          CMPL    R9, #VERIFY$_OPENFILE                       ; 4962
                                      12  13 000D8          BEQL    9$                                          ;
             00000000G   8F           59  D1 000DA          CMPL    R9, #VERIFY$_ENTERLOST                      ; 4963
                                      09  13 000E1          BEQL    9$                                          ;
             00000000G   8F           59  D1 000E3          CMPL    R9, #VERIFY$_DELETE                         ; 4964
                                      0C  12 000EA          BNEQ    10$                                         ;
                  FF74   CD           08  D0 000EC 9$:      MOVL    #8, PARAM                                   ; 4967
                    94   AD      10   AC  D0 000F1          MOVL    EXTRA1, PARAM+32                            ; 4968
                                      3C  11 000F6          BRB     12$                                         ; 4959
             00000000G   8F           59  D1 000F8 10$:     CMPL    R9, #VERIFY$_READFILE                       ; 4971
                                      11  12 000FF          BNEQ    11$                                         ;
                  FF74   CD           09  D0 00101          MOVL    #9, PARAM                                   ; 4974
                  FF7C   CD           06  D0 00106          MOVL    #6, PARAM+8                                 ; 4975
                    94   AD      10   AC  7D 0010B          MOVQ    EXTRA1, PARAM+32                            ; 4976
                                      22  11 00110          BRB     12$                                         ; 4970
```

```
                00000000G  8F          59  D1 00112 11$:   CMPL    R9, #VERIFY$_MULTALLOC          ; 4980
                                       19  12 00119       BNEQ    12$                             ;
                    FF74   CD          0C  D0 0011B       MOVL    #12, PARAM                      ; 4983
                    FF7C   CD          0A  D0 00120       MOVL    #10, PARAM+8                    ; 4984
                      94   AD    10    AC  7D 00125       MOVQ    EXTRA1, PARAM+32                ; 4985
                      9C   AD    18    AC  7D 0012A       MOVQ    EXTRA3, PARAM+40                ; 4987
                      A4   AD    20    AC  D0 0012F       MOVL    EXTRA5, PARAM+48                ; 4989
                00000000G  00    FF74  CD  FA 00134 12$:   CALLG   PARAM, LIB$SIGNAL               ; 4995
                                       04  0013D       RET                                     ; 4996

; Routine Size:  318 bytes,    Routine Base:  CODE + 3A20
```

```
;  5018     4997  1 ROUTINE PROCESS_FILE(NAME,VERSION)=
;  5019     4998  1
;  5020     4999  1 !++
;  5021     5000  1 !
;  5022     5001  1 ! FUNCTIONAL DESCRIPTION:
;  5023     5002  1 !       This routine processes one selected file.
;  5024     5003  1 !
;  5025     5004  1 ! INPUT PARAMETERS:
;  5026     5005  1 !       NAME            - Pointer to ASCIC name string from directory entry.
;  5027     5006  1 !       VERSION         - Pointer to version entry.
;  5028     5007  1 !
;  5029     5008  1 ! IMPLICIT INPUTS:
;  5030     5009  1 !       NONE
;  5031     5010  1 !
;  5032     5011  1 ! OUTPUT PARAMETERS:
;  5033     5012  1 !       NONE
;  5034     5013  1 !
;  5035     5014  1 ! IMPLICIT OUTPUTS:
;  5036     5015  1 !       NONE
;  5037     5016  1 !
;  5038     5017  1 ! ROUTINE VALUE:
;  5039     5018  1 !       True if the directory entry points to a valid file, otherwise false.
;  5040     5019  1 !
;  5041     5020  1 ! SIDE EFFECTS:
;  5042     5021  1 !       NONE
;  5043     5022  1 !
;  5044     5023  1 !--
;  5045     5024  1
;  5046     5025  2 BEGIN
;  5047     5026  2 MAP
;  5048     5027  2         NAME:           REF VECTOR[,BYTE],      ! Pointer to ASCIC name string
;  5049     5028  2         VERSION:        REF BBLOCK;             ! Pointer to version entry
;  5050     5029  2 LOCAL
;  5051     5030  2         FILE_ID:        BBLOCK[FID$C_LENGTH],   ! Clean file ID
;  5052     5031  2         FILE_NUMBER,                            ! Clean file number
;  5053     5032  2         RVN;                                    ! Clean RVN
;  5054     5033  2
;  5055     5034  2
;  5056     5035  2 ! Get clean file ID.
;  5057     5036  2 !
;  5058     5037  2 FILE_NUMBER = .VERSION[DIR$W_FID_NUM];
;  5059     5038  2 FILE_NUMBER<16,8> = .VERSION[DIR$B_FID_NMX];
;  5060     5039  2 RVN = .VERSION[DIR$B_FID_RVN];
;  5061     5040  2 IF .RVN EQL 0 THEN RVN = .DIR_FID[FID$B_RVN];
;  5062     5041  2 FILE_ID[FID$W_NUM] = .FILE_NUMBER;
;  5063     5042  2 FILE_ID[FID$B_NMX] = .FILE_NUMBER<16,8>;
;  5064     5043  2 FILE_ID[FID$W_SEQ] = .VERSION[DIR$W_FID_SEQ];
;  5065     5044  2 FILE_ID[FID$B_RVN] = .RVN;
;  5066     5045  2
;  5067     5046  2
;  5068     5047  2 ! Check file ID.  First, make sure the RVN is in range.  Then, make sure the
;  5069     5048  2 ! file number is in range.  Then, make sure the file number corresponds to a
;  5070     5049  2 ! valid header.  Then, make sure the file sequence number matches that in the
;  5071     5050  2 ! file header.
;  5072     5051  2 !
;  5073     5052  2 IF
;  5074     5053  3     BEGIN
```

```
;  5075          5054  3     IF .RVN GTRU .VOLUME_COUNT
;  5076          5055  3     THEN
;  5077          5056  3         TRUE
;  5078          5057  3     ELSE IF .FILE_NUMBER-1 GTRU .MAXFILIDX[.RVN-1]
;  5079          5058  3     THEN
;  5080          5059  3         TRUE
;  5081          5060  3     ELSE IF NOT .BITVECTOR[.IMAP[.RVN-1], .FILE_NUMBER-1]
;  5082          5061  3     THEN
;  5083          5062  3         TRUE
;  5084          5063  3     ELSE
;  5085          5064  3         .VECTOR[.SEQMAP[.RVN-1], .FILE_NUMBER-1 ;,WORD] NEQ .VERSION[DIR$W_FID_SEQ]
;  5086          5065  3     END
;  5087          5066  2 THEN
;  5088          5067  2     BEGIN
;  5089          5068  2
;  5090          5069  2     ! Invalid file ID.  Report it, and return failure.
;  5091          5070  2     !
;  5092          5071  2     SIGNAL(
;  5093          5072  2         VERIFY$_BADDIRENT,
;  5094          5073  2         3,
;  5095          5074  2         (IF .DIR_DESC[0] EQL 0 THEN MFD_DESC ELSE DIR_DESC),
;  5096          5075  2         .NAME,
;  5097          5076  2         .VERSION[DIR$W_VERSION]);
;  5098          5077  2     IF DO_REPAIR()
;  5099          5078  2     THEN
;  5100          5079  2         ENTER_WORK(WRK_K_REMOVE, FILE_ID, DIR_FID);
;  5101          5080  2     FALSE
;  5102          5081  2     END
;  5103          5082  1 ELSE
;  5104          5083  2     BEGIN
;  5105          5084  2
;  5106          5085  2     ! Check the back link.
;  5107          5086  2     !
;  5108          5087  2     IF .STRUCTURE_LEVEL EQL 2
;  5109          5088  2     THEN
;  5110          5089  3         BEGIN
;  5111          5090  3         LOCAL
;  5112          5091  3             BACK_ID:    REF BBLOCK;
;  5113          5092  3
;  5114          5093  3
;  5115          5094  3         BACK_ID = VECTOR[.BACKMAP[.RVN-1], (.FILE_NUMBER-1)*3 ;,WORD];
;  5116          5095  3
;  5117          5096  3
;  5118          5097  3         ! Compare the back link recorded in the file header to the current
;  5119          5098  3         ! directory file ID.  Note that BACKMAP has clean RVNs.
;  5120          5099  3         !
;  5121          5100  3         IF
;  5122          5101  3             .BACK_ID[FID$W_NUM] NEQ .DIR_FID[FID$W_NUM] OR
;  5123          5102  3             .BACK_ID[FID$W_SEQ] NEQ .DIR_FID[FID$W_SEQ] OR
;  5124          5103  3             .BACK_ID[FID$W_RVN] NEQ .DIR_FID[FID$W_RVN]
;  5125          5104  3         THEN
;  5126          5105  4             BEGIN
;  5127          5106  4
;  5128          5107  4             ! Report incorrect back link.
;  5129          5108  4             !
;  5130          5109  4             SIGNAL(
;  5131          5110  4                 VERIFY$_BACKLINK,
```

```
;  5132          5111  5                3,
;  5133          5112  5                (IF .DIR_DESC[0] EQL 0 THEN MFD_DESC ELSE DIR_DESC),
;  5134          5113  5                .NAME,
;  5135          5114  5                .VERSION[DIR$W_VERSION]);
;  5136          5115  5
;  5137          5116  5
;  5138          5117  5            ! Fix the incorrect back link.
;  5139          5118  5            !
;  5140          5119  5            IF DO_REPAIR() THEN IF READ_HEADER(FILE_ID, BUFFER_2)
;  5141          5120  5            THEN
;  5142          5121  6                BEGIN
;  5143          5122  6                BUFFER_2[FH2$W_BK_FIDNUM] = .DIR_FID[FID$W_NUM];
;  5144          5123  6                BUFFER_2[FH2$W_BK_FIDSEQ] = .DIR_FID[FID$W_SEQ];
;  5145          5124  6                BUFFER_2[FH2$W_BK_FIDRVN] = .DIR_FID[FID$W_RVN];
;  5146          5125  6                IF .DIR_FID[FID$B_RVN] EQL .RVN THEN BUFFER_2[FH2$B_BK_FIDRVN] = 0;
;  5147          5126  6                WRITE_HEADER(FILE_ID, BUFFER_2);
;  5148          5127  5                END;
;  5149          5128  4            END;
;  5150          5129  3        END;
;  5151          5130  3
;  5152          5131  3
;  5153          5132  3        ! Generate usage file entry if requested and if this is the first time
;  5154          5133  3        ! this file has been encountered in the directory scan.
;  5155          5134  3        !
;  5156          5135  3        IF .QUAL[QUAL_USAG] AND .BITVECTOR[.LOSTMAP[.RVN-1], .FILE_NUMBER-1]
;  5157          5136  3        THEN
;  5158          5137  4            BEGIN
;  5159          5138  4            USAGE_BUFFER[USG$B_TYPE] = USG$K_FILE;
;  5160          5139  4            USAGE_BUFFER[USG$L_FILEOWNER] = .VECTOR[.OWNER[.RVN-1], .FILE_NUMBER-1];
;  5161          5140  4            USAGE_BUFFER[USG$L_ALLOCATED] = .VECTOR[.ALLOCATION[.RVN-1], .FILE_NUMBER-1];
;  5162          5141  4            USAGE_BUFFER[USG$L_USED] = .VECTOR[.USAGE[.RVN-1], .FILE_NUMBER-1];
;  5163          5142  4            USAGE_BUFFER[USG$W_DIR_LEN] =
;  5164          5143  4                (IF .DIR_DESC[0] EQL 0 THEN 8 ELSE .DIR_DESC[0] + 2);
;  5165        P 5144  4            $FAO(
;  5166        P 5145  4                $DESCRIPTOR('[!AS]!AC;!UW'),
;  5167        P 5146  4                USAGE_BUFFER[USG$W_SPEC_LEN],
;  5168        P 5147  4                UPLIT(
;  5169        P 5148  4                    %ALLOCATION(USAGE_BUFFER) - $BYTEOFFSET(USG$T_FILESPEC),
;  5170        P 5149  4                    USAGE_BUFFER[USG$T_FILESPEC]),
;  5171        P 5150  4                (IF .DIR_DESC[0] EQL 0 THEN MFD_DESC ELSE DIR_DESC),
;  5172        P 5151  4                .NAME,
;  5173          5152  4                .VERSION[DIR$W_VERSION]);
;  5174          5153  4            USAGE_RAB[RAB$W_RSZ] = $BYTEOFFSET(USG$T_FILESPEC) + .USAGE_BUFFER[USG$W_SPEC_LEN];
;  5175          5154  5            IF NOT $PUT(RAB=USAGE_RAB)
;  5176          5155  4            THEN
;  5177          5156  4                FILE_ERROR(
;  5178          5157  4                    VERIFY$_FACILITY + SHR$_WRITEERR + STS$K_SEVERE,
;  5179          5158  4                    USAGE_FAB,
;  5180          5159  4                    .USAGE_RAB[RAB$L_STS], .USAGE_RAB[RAB$L_STV]);
;  5181          5160  3            END;
;  5182          5161  3
;  5183          5162  3
;  5184          5163  3        ! Mark the file as found in a directory.
;  5185          5164  3        !
;  5186          5165  3        BITVECTOR[.LOSTMAP[.RVN-1], .FILE_NUMBER-1] = FALSE;
;  5187          5166  3        TRUE
;  5188          5167  3        END
```

```
; 5189          5168  1 END;


                57 55 21 38 43 41 21 5D 53 41 21 5B  03B5E P.ABP:  .ASCII  \[!AS]!AC;!UW\                    ;
                                                     03B6A        .BLKB   2                                   ;
                                          0000000C   03B6C P.ABQ:  .LONG   12                                  ;
                                          00000000'  03B70        .ADDRESS P.ABP                              ;
                                          00000196   03B74 P.ABQ:  .LONG   406                                 ;
                                          00000000'  03B78        .ADDRESS USAGE_BUFFER+17                    ;


                                                     00FC 00000 PROCESS_FILE:
                                                                .WORD   Save R2,R3,R4,R5,R6,R7                  ; 4997
                                          57 00000000G   00  9E 00002       MOVAB   LIB$SIGNAL, R7            ;
                                          56     C47F    CF  9E 00009       MOVAB   MFD_DESC, R6              ;
                                          55 00000000'   EF  9E 0000E       MOVAB   DIR_DESC, R5              ;
                                          5E             08  C2 00015       SUBL2   #8, SP                    ;
                                          54       08    AC  D0 00018       MOVL    VERSION, R4               ; 5037
                                          52       02    A4  3C 0001C       MOVZWL  2(R4), FILE_NUMBER        ;
                52                  08    10    07    A4  F0 00020       INSV    7(R4), #16, #8, FILE_NUMBER ; 5038
                                          53       06    A4  9A 00026       MOVZBL  6(R4), RVN                ; 5039
                                                   04  12 0002A       BNEQ    1$                            ; 5040
                                          53       0C    A5  9A 0002C       MOVZBL  DIR_FID+4, RVN            ;
                                          6E             52  B0 00030 1$:    MOVW    FILE_NUMBER, FILE_ID      ; 5041
                50                  52    08    10    EF  00033       EXTZV   #16, #8, FILE_NUMBER, R0  ; 5042
                                  05  AE          50  90 00038       MOVB    R0, FILE_ID+5             ;
                                  02  AE    04    A4  B0 0003C       MOVW    4(R4), FILE_ID+2          ; 5043
                                  04  AE          53  90 00041       MOVB    RVN, FILE_ID+4            ; 5044
                                  20  A5          53  D1 00045       CMPL    RVN, VOLUME_COUNT         ; 5054
                                                   28  1A 00049       BGTRU   2$                        ;
                                          50       FF    A2  9E 0004B       MOVAB   -1(R2), R0                ; 5057
                                          51     7C B543   DE  0004F       MOVAL   @MAXFILIDX[RVN], R1       ;
                                      FC  A1          50  D1 00054       CMPL    R0, -4(R1)                ;
                                                      19  1A 00058       BGTRU   2$                        ;
                                          51   0080 D543   DE  0005A       MOVAL   @IMAP[RVN], R1            ; 5060
                                0E    FC  B1          50  E1 00060       BBC     R0, @-4(R1), 2$           ;
                                          51   0088 D543   DE  00065       MOVAL   @SEQMAP[RVN], R1          ; 5064
                                  04  A4      FC B140   B1  0006B       CMPW    @-4(R1)[R0], 4(R4)        ;
                                                      37  13 00071       BEQL    6$                        ;
                                      7E              64  32 00073 2$:    CVTWL   (R4), -(SP)               ; 5076
                                                  04  AC  DD 00076       PUSHL   NAME                      ; 5075
                                                      65  D5 00079       TSTL    DIR_DESC                  ; 5074
                                                      05  12 0007B       BNEQ    3$                        ;
                                          50              66  9E 0007D       MOVAB   MFD_DESC, R0              ;
                                                      03  11 00080       BRB     4$                        ;
                                          50              65  9E 00082 3$:    MOVAB   DIR_DESC, R0              ;
                                                      50  DD 00085 4$:    PUSHL   R0                        ;
                                                      03  DD 00087       PUSHL   #3                        ; 5071
                                               00000000G  8F  DD 00089       PUSHL   #VERIFY$_BADDIRENT        ;
                                          67              05  FB 0008F       CALLS   #5, LIB$SIGNAL            ;
                                0000V     CF              00  FB 00092       CALLS   #0, DO_REPAIR             ; 5077
                                          0D              50  E9 00097       BLBC    R0, 5$                    ;
                                                  08  A5  9F 0009A       PUSHAB  DIR_FID                   ; 5079
                                                  04  AE  9F 0009D       PUSHAB  FILE_ID                   ;
                                                      01  DD 000A0       PUSHL   #1                        ;
```

```
                     0000V  CF           03  FB 000A2       CALLS   #3, ENTER_WORK
                                         50  D4 000A7 5$:   CLRL    R0                                  ; 5067
                                         04  000A9          RET
                            02     24    A5  D1 000AA 6$:   CMPL    STRUCTURE_LEVEL, #2                 ; 5087
                                         03  13 000AE       BEQL    7$
                                       0080  31 000B0       BRW     12$
                            51     008C D543  DE 000B3 7$:  MOVAL   @BACKMAP[RVN], R1                   ; 5094
                   50       52           03  C5 000B9       MULL3   #3, FILE_NUMBER, R0
                            50       FC B140  3E 000BD      MOVAW   @-4(R1)[R0], BACK_ID
                            50           04  C2 000C2       SUBL2   #4, BACK_ID
                     08     A5           70  B1 000C5       CMPW    -(BACK_ID), DIR_FID                 ; 5101
                                         0E  12 000C9       BNEQ    8$
                     0A     A5     02    A0  B1 000CB       CMPW    2(BACK_ID), DIR_FID+2               ; 5102
                                         07  12 000D0       BNEQ    8$
                     0C     A5     04    A0  B1 000D2       CMPW    4(BACK_ID), DIR_FID+4               ; 5103
                                         5A  13 000D7       BEQL    12$
                            7E           64  32 000D9 8$:   CVTWL   (R4), -(SP)                         ; 5114
                                   04    AC  DD 000DC       PUSHL   NAME                                ; 5113
                                         65  D5 000DF       TSTL    DIR_DESC                            ; 5112
                                         05  12 000E1       BNEQ    9$
                            50           66  9E 000E3       MOVAB   MFD_DESC, R0
                                         03  11 000E6       BRB     10$
                            50           65  9E 000E8 9$:   MOVAB   DIR_DESC, R0
                                         50  DD 000EB 10$:  PUSHL   R0
                                         03  DD 000ED       PUSHL   #3                                  ; 5109
                              00000000G  8F  DD 000EF       PUSHL   #VERIFY$_BACKLINK
                            67           05  FB 000F5       CALLS   #5, LIB$SIGNAL
                     0000V  CF           00  FB 000F8       CALLS   #0, DO_REPAIR                       ; 5119
                            33           50  E9 000FD       BLBC    R0, 12$
                                   FC5C  C5  9F 00100       PUSHAB  BUFFER_2
                                     04  AE  9F 00104       PUSHAB  FILE_ID
                     F99E   CF           02  FB 00107       CALLS   #2, READ_HEADER
                            24           50  E9 0010C       BLBC    R0, 12$
                     FC9E   C5     08    A5  D0 0010F       MOVL    DIR_FID, BUFFER_2+66                ; 5122
                     FCA2   C5     0C    A5  B0 00115       MOVW    DIR_FID+4, BUFFER_2+70              ; 5124
  53        0C      A5      08           00  ED 0011B       CMPZV   #0, #8, DIR_FID+4, RVN              ; 5125
                                         04  12 00121       BNEQ    11$
                                   FCA2  C5  94 00123       CLRB    BUFFER_2+70
                                   FC5C  C5  9F 00127 11$:  PUSHAB  BUFFER_2                            ; 5126
                                     04  AE  9F 0012B       PUSHAB  FILE_ID
                     FA1D   CF           02  FB 0012E       CALLS   #2, WRITE_HEADER
             03 00000000'   EF           04  E0 00133 12$:  BBS     #4, QUAL, 14$                       ; 5135
                                       00C5  31 0013B 13$:  BRW     19$
                            50     0090 D543  DE 0013E 14$: MOVAL   @LOSTMAP[RVN], R0
                            51     FF    A2  9E 00144       MOVAB   -1(R2), R1
             EE        FC   B0           51  E1 00148       BBC     R1, @-4(R0), 13$
                00000000'   EF           02  90 0014D       MOVB    #2, USAGE_BUFFER                    ; 5138
                            50     0098 D543  DE 00154      MOVAL   @OWNER[RVN], R0                     ; 5139
                            50       FC B042  DE 0015A      MOVAL   @-4(R0)[FILE_NUMBER], R0
                00000000'   EF     FC    A0  D0 0015F       MOVL    -4(R0), USAGE_BUFFER+1
                            50     009C D543  DE 00167      MOVAL   @ALLOCATION[RVN], R0                ; 5140
                            50       FC B042  DE 0016D      MOVAL   @-4(R0)[FILE_NUMBER], R0
                00000000'   EF     FC    A0  D0 00172       MOVL    -4(R0), USAGE_BUFFER+5
                            50     00A0 D543  DE 0017A      MOVAL   @USAGE[RVN], R0                     ; 5141
                            50       FC B042  DE 00180      MOVAL   @-4(R0)[FILE_NUMBER], R0
                00000000'   EF     FC    A0  D0 00185       MOVL    -4(R0), USAGE_BUFFER+9
                            50           65  D0 0018D       MOVL    DIR_DESC, R0                        ; 5143
```

```
                                                51  D4 00190          CLRL    R1                                                          ;
                                                50  D5 00192          TSTL    R0                                                          ;
                                                07  12 00194          BNEQ    15$                                                         ;
                                                51  D6 00196          INCL    R1                                                          ;
                                    50          08  D0 00198          MOVL    #8, R0                                                      ;
                                                03  11 0019B          BRB     16$                                                         ;
                                    50          02  C0 0019D 15$:     ADDL2   #2, R0                                                      ;
                        00000000'   EF          50  B0 001A0 16$:     MOVW    R0, USAGE_BUFFER+13                                         ;
                                    7E              64  32 001A7      CVTWL   (R4), -(SP)                                                 ; 5152
                                          04    AC  DD 001AA          PUSHL   NAME                                                        ;
                                    05          51  E9 001AD          BLBC    R1, 17$                                                     ;
                                    50          66  9E 001B0          MOVAB   MFD_DESC, R0                                                ;
                                                03  11 001B3          BRB     18$                                                         ;
                                    50          65  9E 001B5 17$:     MOVAB   DIR_DESC, R0                                                ;
                                                50  DD 001B8 18$:     PUSHL   R0                                                          ;
                                          3B6C  C6  9F 001BA          PUSHAB  P.ABQ                                                       ;
                                00000000'       EF  9F 001BE          PUSHAB  USAGE_BUFFER+15                                             ;
                                          3B64  C6  9F 001C4          PUSHAB  P.ABO                                                       ;
                        00000000G   00          06  FB 001C8          CALLS   #6, SYS$FAO                                                 ;
00000000'  EF   00000000'   EF                  11  A1 001CF          ADDW3   #17, USAGE_BUFFER+15, USAGE_RAB+34                          ; 5153
                                00000000'       EF  9F 001DB          PUSHAB  USAGE_RAB                                                   ; 5154
                        00000000G   00          01  FB 001E1          CALLS   #1, SYS$PUT                                                 ;
                                    18          50  E8 001E8          BLBS    R0, 19$                                                     ;
                                    7E  00000000'   EF  7D 001EB      MOVQ    USAGE_RAB+8, -(SP)                                          ; 5159
                                00000000'       EF  9F 001F2          PUSHAB  USAGE_FAB                                                   ; 5156
                                00000000G       8F  DD 001F8          PUSHL   #VERIFY$_FACILITY+4308                                      ; 5157
                            FC34    CF          04  FB 001FE          CALLS   #4, FILE_ERROR                                              ;
                                    50      0090 D543   DE 00203 19$: MOVAL   @LOSTMAP[RVN], R0                                          ; 5165
                                                52  D7 00209          DECL    R2                                                          ;
                        00      FC  B0          52  E5 0020B          BBCC    R2, @-4(R0), 20$                                            ;
                                    50          01  D0 00210 20$:     MOVL    #1, R0                                                      ; 5083
                                                    04 00213          RET                                                                 ; 5168

; Routine Size:  532 bytes,    Routine Base:  CODE + 3B7C
```

```
;  5191     5169  1 ROUTINE PROCESS_SUBDIR(NAME,VERSION): NOVALUE=
;  5192     5170  1
;  5193     5171  1 !++
;  5194     5172  1 !
;  5195     5173  1 ! FUNCTIONAL DESCRIPTION:
;  5196     5174  1 !       This routine processes one subdirectory entry.
;  5197     5175  1 !
;  5198     5176  1 ! INPUT PARAMETERS:
;  5199     5177  1 !       NAME            - Pointer to ASCIC name string from directory entry.
;  5200     5178  1 !       VERSION         - Pointer to version entry.
;  5201     5179  1 !
;  5202     5180  1 ! IMPLICIT INPUTS:
;  5203     5181  1 !       NONE
;  5204     5182  1 !
;  5205     5183  1 ! OUTPUT PARAMETERS:
;  5206     5184  1 !       NONE
;  5207     5185  1 !
;  5208     5186  1 ! IMPLICIT OUTPUTS:
;  5209     5187  1 !       NONE
;  5210     5188  1 !
;  5211     5189  1 ! ROUTINE VALUE:
;  5212     5190  1 !       NONE
;  5213     5191  1 !
;  5214     5192  1 ! SIDE EFFECTS:
;  5215     5193  1 !       NONE
;  5216     5194  1 !
;  5217     5195  1 !--
;  5218     5196  1
;  5219     5197  2 BEGIN
;  5220     5198  2 MAP
;  5221     5199  2         NAME:           REF VECTOR[,BYTE],      ! Pointer to ASCIC name string
;  5222     5200  2         VERSION:        REF BBLOCK;             ! Pointer to version entry
;  5223     5201  2 LOCAL
;  5224     5202  2         SAVE_DIR_DESC,                          ! Recursive save for directory string length
;  5225     5203  2         SAVE_DIR_FID:   BBLOCK[FID$C_LENGTH],   ! Recursive save area for directory file ID
;  5226     5204  2         STATUS,                                 ! Status variable
;  5227     5205  2         DIR_LENGTH,                             ! Length of directory
;  5228     5206  2         BUF_LENGTH,                             ! Length of buffer for directory
;  5229     5207  2         BUF_ADDRESS;                            ! Address of buffer for directory
;  5230     5208  2
;  5231     5209  2
;  5232     5210  2 ! Save recursive variables.
;  5233     5211  2 !
;  5234     5212  2 SAVE_DIR_DESC = .DIR_DESC[0];
;  5235     5213  2 SAVE_DIR_FID[FID$W_NUM] = .DIR_FID[FID$W_NUM];
;  5236     5214  2 SAVE_DIR_FID[FID$W_SEQ] = .DIR_FID[FID$W_SEQ];
;  5237     5215  2 SAVE_DIR_FID[FID$W_RVN] = .DIR_FID[FID$W_RVN];
;  5238     5216  2
;  5239     5217  2
;  5240     5218  2 ! Update the directory descriptor, unless NAME is zero, which indicates MFD.
;  5241     5219  2 ! If this is not the top level, append a dot, and then append the directory
;  5242     5220  2 ! name, up to but not including the dot.
;  5243     5221  2 !
;  5244     5222  2 IF .NAME NEQ 0
;  5245     5223  2 THEN
;  5246     5224  3     BEGIN
;  5247     5225  3     LOCAL
```

```
;  5248     5226  3          P;
;  5249     5227
;  5250     5228  3      IF .DIR_DESC[0] NEQ 0
;  5251     5229  3      THEN
;  5252     5230  4          BEGIN
;  5253     5231  4          DIR[.DIR_DESC[0]] = %C'.';
;  5254     5232  4          DIR_DESC[0] = .DIR_DESC[0] + 1;
;  5255     5233  4          END;
;  5256     5234
;  5257     5235  3      P = CH$FIND_CH(.NAME[0], NAME[1], %C'.');
;  5258     5236  3      IF .P NEQ 0
;  5259     5237  3      THEN
;  5260     5238  3          DIR_DESC[0] =
;  5261     5239  3              CH$MOVE(.P - NAME[1], NAME[1], DIR[.DIR_DESC[0]]) - .DIR_DESC[1];
;  5262     5240  2      END;
;  5263     5241  2
;  5264     5242  2
;  5265     5243  2  ! Access the file.
;  5266     5244  2  !
;  5267     5245  2  CH$FILL(0, FIB$C_LENGTH, FIB);
;  5268     5246  2  FIB[FIB$L_ACCTL] = FIB$M_NORECORD;
;  5269     5247  2  FIB[FIB$W_FID_NUM] = .VERSION[DIR$W_FID_NUM];
;  5270     5248  2  FIB[FIB$W_FID_SEQ] = .VERSION[DIR$W_FID_SEQ];
;  5271     5249  2  FIB[FIB$W_FID_RVN] = .VERSION[DIR$W_FID_RVN];
;  5272     5250  2  IF .FIB[FIB$B_FID_RVN] EQL 0 THEN FIB[FIB$B_FID_RVN] = .DIR_FID[FID$B_RVN];
;  5273     5251  2  DIR_FID[FID$W_NUM] = .FIB[FIB$W_FID_NUM];
;  5274     5252  2  DIR_FID[FID$W_SEQ] = .FIB[FIB$W_FID_SEQ];
;  5275     5253  2  DIR_FID[FID$W_RVN] = .FIB[FIB$W_FID_RVN];
;  5276   P 5254  2  STATUS = $QIOW(
;  5277   P 5255  2      FUNC=IO$_ACCESS OR IO$M_ACCESS,
;  5278   P 5256  2      CHAN=.CHANNEL,
;  5279   P 5257  2      IOSB=IOSB,
;  5280   P 5258  2      P1=FIB_DESC,
;  5281     5259  2      P5=HDR_ATR_DESC);
;  5282     5260  2  IF .STATUS THEN STATUS = .IOSB[0];
;  5283     5261  2  IF NOT .STATUS
;  5284     5262  2  THEN
;  5285     5263  3      BEGIN
;  5286     5264  3
;  5287     5265  3      ! Report failure to access the directory, and set the error flag to
;  5288     5266  3      ! abort lost file processing.
;  5289     5267  3      !
;  5290     5268  3      DIRECTORY_ERROR = TRUE;
;  5291     5269  3      SIGNAL(
;  5292     5270  3          VERIFY$_OPENDIR,
;  5293     5271  3          1,
;  5294     5272  3          (IF .DIR_DESC[0] EQL 0 THEN MFD_DESC ELSE DIR_DESC),
;  5295     5273  3          .STATUS);
;  5296     5274  3      END
;  5297     5275  2  ELSE
;  5298     5276  3      BEGIN
;  5299     5277  3
;  5300     5278  3      ! Ensure that the file is, in fact, a directory.  At this point we know
;  5301     5279  3      ! only that the filename is ".DIR;1".  Use the file header that was
;  5302     5280  3      ! obtained during the access.  Compute the file length if valid.
;  5303     5281  3      ! If invalid, leave the file length zero to avoid processing the file.
;  5304     5282  3      !
```

```
;  5305     5283  3      DIR_LENGTH = 0;
;  5306     5284  3      IF .HDR_BUFFER[FH2$B_STRUCLEV] EQL 2
;  5307     5285  3      THEN
;  5308     5286  4          BEGIN
;  5309     5287  4          BIND
;  5310     5288  4              RECATTR=    HDR_BUFFER[FH2$W_RECATTR]: BBLOCK;
;  5311     5289  4
;  5312     5290  4          IF .HDR_BUFFER[FH2$V_DIRECTORY]
;  5313     5291  4          THEN
;  5314     5292  5              BEGIN
;  5315     5293  5              DIR_LENGTH = ROT(.RECATTR[FAT$L_EFBLK], 16) * 512;
;  5316     5294  5              IF .RECATTR[FAT$W_FFBYTE] EQL 0 THEN DIR_LENGTH = .DIR_LENGTH - 512;
;  5317     5295  4              END;
;  5318     5296  4          END
;  5319     5297  3      ELSE
;  5320     5298  4          BEGIN
;  5321     5299  4          BIND
;  5322     5300  4              RECATTR=    HDR_BUFFER[FH1$W_RECATTR]: BBLOCK;
;  5323     5301  4
;  5324     5302  4          IF
;  5325     5303  4              .RECATTR[FAT$B_RTYPE] EQL FAT$C_FIXED AND
;  5326     5304  4              .RECATTR[FAT$W_RSIZE] EQL NMB$C_DIRENTRY
;  5327     5305  4          THEN
;  5328     5306  5              BEGIN
;  5329     5307  5              DIR_LENGTH = ROT(.RECATTR[FAT$L_EFBLK], 16) * 512;
;  5330     5308  5              IF .RECATTR[FAT$W_FFBYTE] EQL 0 THEN DIR_LENGTH = .DIR_LENGTH - 512;
;  5331     5309  4              END;
;  5332     5310  3          END;
;  5333     5311  3
;  5334     5312  3
;  5335     5313  3      ! Check for proper end of file pointer.
;  5336     5314  3      !
;  5337     5315  3      IF
;  5338     5316  3          .DIR_LENGTH LSS 0 OR
;  5339     5317  4          (.DIR_LENGTH EQL 0 AND .HDR_BUFFER[FH2$B_STRUCLEV] EQL 2)
;  5340     5318  3      THEN
;  5341     5319  4          BEGIN
;  5342     5320  4          DIRECTORY_ERROR = TRUE;
;  5343     5321  4          SIGNAL(
;  5344     5322  4              VERIFY$_BADDIR,
;  5345     5323  4              1,
;  5346     5324  4              (IF .DIR_DESC[0] EQL 0 THEN MFD_DESC ELSE DIR_DESC));
;  5347     5325  4          DIR_LENGTH = 0;
;  5348     5326  3          END;
;  5349     5327  3
;  5350     5328  3
;  5351     5329  3      IF .DIR_LENGTH NEQ 0
;  5352     5330  3      THEN
;  5353     5331  4          BEGIN
;  5354     5332  4          LOCAL
;  5355     5333  4              READ_VBN;
;  5356     5334  4
;  5357     5335  4
;  5358     5336  4          ! Compute buffer length.  Try to read the entire directory at
;  5359     5337  4          ! one time, but no more than DIR_BUF_COUNT blocks.
;  5360     5338  4          !
;  5361     5339  4          BUF_LENGTH = MINU(DIR_BUF_COUNT*512, .DIR_LENGTH);
```

```
;  5362          5340  4
;  5363          5341  4
;  5364          5342  4          ! Allocate memory for buffer.
;  5365          5343  4          !
;  5366          5344  4          STATUS = LIB$GET_VM(BUF_LENGTH, BUF_ADDRESS);
;  5367          5345  4          IF NOT .STATUS THEN SIGNAL(VERIFY$_ALLOCMEM, 0, .STATUS);
;  5368          5346  4
;  5369          5347  4
;  5370          5348  4          ! Loop to process all blocks.
;  5371          5349  4          !
;  5372          5350  4          READ_VBN = 1;
;  5373          5351  4          WHILE 1 DO
;  5374          5352  5              BEGIN
;  5375          5353  5              LOCAL
;  5376          5354  5                  PROC_LENGTH;
;  5377          5355  5
;  5378          5356  5
;  5379          5357  5              ! Compute transfer size.
;  5380          5358  5              !
;  5381          5359  5              PROC_LENGTH = MINU(.BUF_LENGTH, (.DIR_LENGTH/512 - .READ_VBN + 1)*512);
;  5382          5360  5
;  5383          5361  5
;  5384          5362  5              ! Read the blocks.
;  5385          5363  5              !
;  5386        P 5364  5              STATUS = $QIOW(
;  5387        P 5365  5                  FUNC=IO$_READVBLK,
;  5388        P 5366  5                  CHAN=.CHANNEL,
;  5389        P 5367  5                  IOSB=IOSB,
;  5390        P 5368  5                  P1=.BUF_ADDRESS,
;  5391        P 5369  5                  P2=.PROC_LENGTH,
;  5392          5370  5                  P3=.READ_VBN);
;  5393          5371  5              IF .STATUS THEN STATUS = .IOSB[0];
;  5394          5372  5              IF NOT .STATUS
;  5395          5373  5              THEN
;  5396          5374  5                  SIGNAL(
;  5397          5375  5                      VERIFY$_READDIR,
;  5398          5376  5                      1,
;  5399          5377  5                      (IF .DIR_DESC[0] EQL 0 THEN MFD_DESC ELSE DIR_DESC),
;  5400          5378  5                      .STATUS);
;  5401          5379  5
;  5402          5380  5
;  5403          5381  5              ! Deaccess the file.
;  5404          5382  5              !
;  5405        P 5383  5              $QIOW(
;  5406        P 5384  5                  FUNC=IO$_DEACCESS,
;  5407          5385  5                  CHAN=.CHANNEL);
;  5408          5386  5
;  5409          5387  5
;  5410          5388  5              ! Scan directory, if there was no error in the read.
;  5411          5389  5              !
;  5412          5390  5              IF .STATUS
;  5413          5391  5              THEN
;  5414          5392  6                  BEGIN
;  5415          5393  7                  (IF .STRUCTURE_LEVEL EQL 2
;  5416          5394  7                      THEN SCAN_DIRECT_2
;  5417          5395  7                      ELSE SCAN_DIRECT_1)
;  5418          5396  6                  (.PROC_LENGTH, .BUF_ADDRESS)
```

```
;  5419     5397  6                END
;  5420     5398  5            ELSE
;  5421     5399  6                BEGIN
;  5422     5400  6                DIRECTORY_ERROR = TRUE;
;  5423     5401  6                EXITLOOP;
;  5424     5402  6                END;
;  5425     5403  5
;  5426     5404  5
;  5427     5405  5            ! Update to next chunk.
;  5428     5406  5            !
;  5429     5407  5            READ_VBN = .READ_VBN + .PROC_LENGTH/512;
;  5430     5408  5            IF .READ_VBN GTRU .DIR_LENGTH/512 THEN EXITLOOP;
;  5431     5409  5
;  5432     5410  5
;  5433     5411  5            ! Re-access the file for another trip.
;  5434     5412  5            !
;  5435     5413  5            CH$FILL(0, FIB$C_LENGTH, FIB);
;  5436     5414  5            FIB[FIB$L_ACCTL] = FIB$M_NORECORD;
;  5437     5415  5            FIB[FIB$W_FID_NUM] = .DIR_FID[FID$W_NUM];
;  5438     5416  5            FIB[FIB$W_FID_SEQ] = .DIR_FID[FID$W_SEQ];
;  5439     5417  5            FIB[FIB$W_FID_RVN] = .DIR_FID[FID$W_RVN];
;  5440   P 5418  5            STATUS = $QIOW(
;  5441   P 5419  5                FUNC=IO$_ACCESS OR IO$M_ACCESS,
;  5442   P 5420  5                CHAN=.CHANNEL,
;  5443   P 5421  5                IOSB=IOSB,
;  5444     5422  5                P1=FIB_DESC);
;  5445     5423  5            IF .STATUS THEN STATUS = .IOSB[0];
;  5446     5424  5            IF NOT .STATUS
;  5447     5425  5            THEN
;  5448     5426  6                BEGIN
;  5449     5427  6
;  5450     5428  6                ! Report failure to access the directory, and set the error flag to
;  5451     5429  6                ! abort lost file processing.
;  5452     5430  6                !
;  5453     5431  6                DIRECTORY_ERROR = TRUE;
;  5454     5432  6                SIGNAL(
;  5455     5433  6                    VERIFY$_OPENDIR,
;  5456     5434  6                    1,
;  5457     5435  6                    (IF .DIR_DESC[0] EQL 0 THEN MFD_DESC ELSE DIR_DESC),
;  5458     5436  6                    .STATUS);
;  5459     5437  6                EXITLOOP;
;  5460     5438  5                END;
;  5461     5439  4            END;
;  5462     5440  4
;  5463     5441  4
;  5464     5442  4            ! Deallocate memory for working copy of directory.
;  5465     5443  4            !
;  5466     5444  4            STATUS = LIB$FREE_VM(BUF_LENGTH, BUF_ADDRESS);
;  5467     5445  4            IF NOT .STATUS THEN SIGNAL(VERIFY$_FREEMEM, 0, .STATUS);
;  5468     5446  4            END
;  5469     5447  3        ELSE
;  5470     5448  4            BEGIN
;  5471     5449  4
;  5472     5450  4            ! Deaccess the file.
;  5473     5451  4            !
;  5474   P 5452  4            $QIOW(
;  5475   P 5453  4                FUNC=IO$_DEACCESS,
```

```
;  5476     5454  4              CHAN=.CHANNEL);
;  5477     5455  3          END;
;  5478     5456  2      END;
;  5479     5457  2
;  5480     5458  2
;  5481     5459  2  ! Restore recursive variables.
;  5482     5460  2  !
;  5483     5461  2  DIR_DESC[0] = .SAVE_DIR_DESC;
;  5484     5462  2  DIR_FID[FID$W_NUM] = .SAVE_DIR_FID[FID$W_NUM];
;  5485     5463  2  DIR_FID[FID$W_SEQ] = .SAVE_DIR_FID[FID$W_SEQ];
;  5486     5464  2  DIR_FID[FID$W_RVN] = .SAVE_DIR_FID[FID$W_RVN];
;  5487     5465  1  END;


                                                                RECATTR=            BUFFER+532
                                                                RECATTR=            BUFFER+526


                                                    OFFC 00000 PROCESS_SUBDIR:
                                                               .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11        ; 5169
                                                    5B 00000000G   00  9E 00002     MOVAB   LIB$SIGNAL, R11
                                                    5A 00000000'   EF  9E 00009     MOVAB   DIR_DESC, R10
                                                    5E             10  C2 00010     SUBL2   #16, SP
                                                    50             6A  D0 00013     MOVL    DIR_DESC, R0                ; 5212
                                                    59             50  D0 00016     MOVL    R0, SAVE_DIR_DESC
                                              08    AE       08    AA  D0 00019     MOVL    DIR_FID, SAVE_DIR_FID       ; 5213
                                              0C    AE       0C    AA  B0 0001E     MOVW    DIR_FID+4, SAVE_DIR_FID+4   ; 5215
                                                    52       04    AC  D0 00023     MOVL    NAME, R2                    ; 5222
                                                                   34  13 00027     BEQL    3$
                                                                   50  D5 00029     TSTL    R0                          ; 5228
                                                                   08  13 0002B     BEQL    1$
                                                             FEC0  CA40            2E  90 0002D     MOVB    #46, DIR[R0]                ; 5231
                                                                   6A  D6 00033     INCL    DIR_DESC                    ; 5232
                                                                50             62  9A 00035 1$:     MOVZBL  (R2)+, R0           ; 5235
                                                          01    A2       50      2E  3A 00038     LOCC    #46, R0, 1(R2)
                                                                   02  12 0003D     BNEQ    2$
                                                                   51  D4 0003F     CLRL    R1
                                                                   51  D5 00041 2$:     TSTL    P                       ; 5236
                                                                   18  13 00043     BEQL    3$
                                                                50       01    A2  9E 00045     MOVAB   1(R2), R0           ; 5239
                                                                51             50  C2 00049     SUBL2   R0, R1
                                                                50       FEC0  CA  9E 0004C     MOVAB   DIR, R0
                                                          00 BA40        01    A2  28 00051     MOVC3   R1, 1(R2), @DIR_DESC[R0]
                                                                6A             53  C3 00058     SUBL3   DIR_DESC+4, R3, DIR_DESC
                                                    0040  8F              00    2C 0005D 3$:     MOVC5   #0, (SP), #0, #64, FIB  ; 5245
                                                                   FE5C  CA      00064
                                                             FE5C  CA 00200000  8F  D0 00067     MOVL    #2097152, FIB       ; 5246
                                                                50             08  AC  D0 00070     MOVL    VERSION, R0         ; 5247
                                                             FE60  CA          02  A0  D0 00074     MOVL    2(R0), FIB+4
                                                             FE64  CA          06  A0  B0 0007A     MOVW    6(R0), FIB+8        ; 5249
                                                                   FE64  CA        95  00080     TSTB    FIB+8               ; 5250
                                                                         06  12 00084     BNEQ    4$
                                                             FE64  CA          0C  AA  90 00086     MOVB    DIR_FID+4, FIB+8
                                                                08  AA       FE60  CA  D0 0008C 4$:     MOVL    FIB+4, DIR_FID      ; 5251
                                                                0C  AA       FE64  CA  B0 00092     MOVW    FIB+8, DIR_FID+4    ; 5253
                                                                                   7E  D4 00098     CLRL    -(SP)               ; 5259
```

```
                                 C1F6    CF  9F 0009A          PUSHAB  HDR_ATR_DESC                                    ;
                                         7E  7C 0009E          CLRQ    -(SP)                                           ;
                                         7E  D4 000A0          CLRL    -(SP)                                           ;
                                 C1FA    CF  9F 000A2          PUSHAB  FIB_DESC                                        ;
                                         7E  7C 000A6          CLRQ    -(SP)                                           ;
                                   18    AA  9F 000A8          PUSHAB  IOSB                                            ;
                              7E   72    8F  9A 000AB          MOVZBL  #114, -(SP)                                     ;
                                 00000000'EF  DD 000AF          PUSHL   CHANNEL                                         ;
                                         7E  D4 000B5          CLRL    -(SP)                                           ;
                        00000000G     00    FB 000B7          CALLS   #12, SYS$QIOW                                   ;
                                      56    50 000BE          MOVL    R0, STATUS                                      ;
                                      07    E9 000C1          BLBC    STATUS, 5$                                      ; 5260
                                      56 18 AA 3C 000C4          MOVZWL  IOSB, STATUS                                    ;
                                      27    E8 000C8          BLBS    STATUS, 8$                                      ; 5261
                        00000000'     EF    D0 000CB 5$:      MOVL    #1, DIRECTORY_ERROR                             ; 5268
                                      56    DD 000D2          PUSHL   STATUS                                          ; 5273
                                      6A    D5 000D4          TSTL    DIR_DESC                                        ; 5272
                                      07    12 000D6          BNEQ    6$                                              ;
                                 50   C19C  CF  9E 000D8          MOVAB   MFD_DESC, R0                                    ;
                                      03    11 000DD          BRB     7$                                              ;
                                 50         6A  9E 000DF 6$:       MOVAB   DIR_DESC, R0                                    ;
                                      50    DD 000E2 7$:      PUSHL   R0                                              ;
                                      01    DD 000E4          PUSHL   #1                                              ; 5269
                                 00000000G  8F  DD 000E6          PUSHL   #VERIFY$_OPENDIR                                ;
                                 6B         04  FB 000EC          CALLS   #4, LIB$SIGNAL                                  ;
                                          022F 31 000EF          BRW     34$                                             ; 5261
                                      52    D4 000F2 8$:      CLRL    DIR_LENGTH                                      ; 5283
                                      51    D4 000F4          CLRL    R1                                              ; 5284
                                 02 00000000' EF  91 000F6          CMPB    HDR_BUFFER+7, #2                                ;
                                      1E    12 000FD          BNEQ    9$                                              ;
                                      51    D6 000FF          INCL    R1                                              ;
                  3F 00000000'        EF    05 E1 00101          BBC     #5, HDR_BUFFER+53, 11$                          ; 5290
                  50 00000000'        EF    10 9C 00109          ROTL    #16, RECATTR+8, R0                              ; 5293
                  52                  50    09 78 00111          ASHL    #9, R0, DIR_LENGTH                              ;
                                 00000000'  CF  B5 00115          TSTW    RECATTR+12                                      ; 5294
                                      24    11 0011B          BRB     10$                                             ;
                                 01 00000000' EF  91 0011D 9$:       CMPB    RECATTR, #1                                     ; 5303
                                      22    12 00124          BNEQ    11$                                             ;
                                 10 00000000' EF  B1 00126          CMPW    RECATTR+2, #16                                  ; 5304
                                      19    12 0012D          BNEQ    11$                                             ;
                  50 00000000'        EF    10 9C 0012F          ROTL    #16, RECATTR+8, R0                              ; 5307
                  52                  50    09 78 00137          ASHL    #9, R0, DIR_LENGTH                              ;
                                 00000000'  EF  B5 0013B          TSTW    RECATTR+12                                      ; 5308
                                      05    12 00141 10$:      BNEQ    11$                                             ;
                                 52   FE00  C2  9E 00143          MOVAB   -512(R2), DIR_LENGTH                            ;
                                      52    D5 00148 11$:      TSTL    DIR_LENGTH                                      ; 5316
                                      05    19 0014A          BLSS    12$                                             ;
                                      2C    12 0014C          BNEQ    16$                                             ; 5317
                                      24    51 E9 0014E          BLBC    R1, 15$                                         ;
                        00000000'     EF    01 D0 00151 12$:      MOVL    #1, DIRECTORY_ERROR                             ; 5320
                                      6A    D5 00158          TSTL    DIR_DESC                                        ; 5324
                                      07    12 0015A          BNEQ    13$                                             ;
                                 50   C118  CF  9E 0015C          MOVAB   MFD_DESC, R0                                    ;
                                      03    11 00161          BRB     14$                                             ;
                                 50         6A  9E 00163 13$:      MOVAB   DIR_DESC, R0                                    ;
                                      50    DD 00166 14$:      PUSHL   R0                                              ;
                                      01    DD 00168          PUSHL   #1                                              ; 5321
```

```
                        00000000G   8F   DD 0016A          PUSHL   #VERIFY$_BADDIR                          ;
                              6B        03   FB 00170          CALLS   #3, LIB$SIGNAL                          ;
                                        52   D4 00173          CLRL    DIR_LENGTH                              ; 5325
                                        03   12 00175 15$:     BNEQ    16$                                     ; 5329
                                      018D   31 00177          BRW     33$                                     ;
                                        50   D0 0017A 16$:     MOVL    DIR_LENGTH, R0                          ; 5339
                          00002000      8F   D1 0017D          CMPL    R0, #8192                               ;
                                        05   1B 00184          BLEQU   17$                                     ;
                                        50     2000   8F   3C 00186          MOVZWL  #8192, R0                       ;
                                04      AE        50   D0 00188B 17$:     MOVL    R0, BUF_LENGTH                  ;
                                        5E   DD 0018F          PUSHL   SP                                      ; 5344
                                          08    AE   9F 00191          PUSHAB  BUF_LENGTH                      ;
                          00000000G     00        02   FB 00194          CALLS   #2, LIB$GET_VM                  ;
                                        56   50   D0 0019B          MOVL    R0, STATUS                      ;
                                        0D        56   E8 0019E          BLBS    STATUS, 18$                     ; 5345
                                        56   DD 001A1          PUSHL   STATUS                          ;
                                        7E   D4 001A3          CLRL    -(SP)                           ;
                                00000000G   8F   DD 001A5          PUSHL   #VERIFY$_ALLOCMEM               ;
                                      6B        03   FB 001AB          CALLS   #3, LIB$SIGNAL                  ;
                                        57   01   D0 001AE 18$:     MOVL    #1, READ_VBN                    ; 5350
                                52      58   00000200   8F   C7 001B1          DIVL3   #512, DIR_LENGTH, R8            ; 5359
                                52      58        57   C3 001B9 19$:     SUBL3   READ_VBN, R8, R2                ;
                                52      52        09   78 001BD          ASHL    #9, R2, R2                      ;
                                        52     0200   C2   9E 001C1          MOVAB   512(R2), R2                     ;
                                        50       04   AE   D0 001C6          MOVL    BUF_LENGTH, R0                  ;
                                        52        50   D1 001CA          CMPL    R0, R2                          ;
                                        03   1B 001CD          BLEQU   20$                             ;
                                        50        52   D0 001CF          MOVL    R2, R0                  ;
                                        52        50   D0 001D2 20$:     MOVL    R0, PROC_LENGTH         ;
                                        7E   7C 001D5          CLRQ    -(SP)                           ; 5370
                                        7E   D4 001D7          CLRL    -(SP)                           ;
                                      0084   8F   BB 001D9          PUSHR   #^M<R2,R7>                      ;
                                        14   AE   DD 001DD          PUSHL   BUF_ADDRESS                     ;
                                        7E   7C 001E0          CLRQ    -(SP)                           ;
                                        18   AA   9F 001E2          PUSHAB  IOSB                            ;
                                        31   DD 001E5          PUSHL   #49                             ;
                                00000000'   EF   DD 001E7          PUSHL   CHANNEL                         ;
                                        7E   D4 001ED          CLRL    -(SP)                           ;
                          00000000G     00        0C   FB 001EF          CALLS   #12, SYS$QIOW                   ;
                                        56   50   D0 001F6          MOVL    R0, STATUS                      ;
                                        07        56   E9 001F9          BLBC    STATUS, 21$                     ; 5371
                                        56     18   AA   3C 001FC          MOVZWL  IOSB, STATUS                    ;
                                        1D        56   E8 00200          BLBS    STATUS, 24$                     ; 5372
                                        56   DD 00203 21$:     PUSHL   STATUS                          ; 5378
                                        6A   D5 00205          TSTL    DIR_DESC                        ; 5377
                                        07   12 00207          BNEQ    22$                             ;
                                50      C06B   CF   9E 00209          MOVAB   MFD_DESC, R0                    ;
                                        03   11 0020E          BRB     23$                             ;
                                50        6A   9E 00210 22$:     MOVAB   DIR_DESC, R0                    ;
                                        50   DD 00213 23$:     PUSHL   R0                              ;
                                        01   DD 00215          PUSHL   #1                              ; 5374
                                00000000G   8F   DD 00217          PUSHL   #VERIFY$_READDIR                ;
                                      6B        04   FB 0021D          CALLS   #4, LIB$SIGNAL                  ;
                                        7E   7C 00220 24$:     CLRQ    -(SP)                           ; 5385
                                        7E   7C 00222          CLRQ    -(SP)                           ;
                                        7E   7C 00224          CLRQ    -(SP)                           ;
                                        7E   7C 00226          CLRQ    -(SP)                           ;
```

```
                            7E               34  7D 00228         MOVQ    #52, -(SP)                     ;
                                00000000'    EF  DD 0022B         PUSHL   CHANNEL                        ;
                                             7E  D4 00231         CLRL    -(SP)                          ;
                 00000000G  00               0C  FB 00233         CALLS   #12, SYS$QIOW                  ;
                            1B               56  E9 0023A         BLBC    STATUS, 27$                    ; 5390
                            02         24    AA  D1 0023D         CMPL    STRUCTURE_LEVEL, #2            ; 5393
                                             07  12 00241         BNEQ    25$                            ;
                            50      0000V    CF  9E 00243         MOVAB   SCAN_DIRECT_2, R0              ;
                                             05  11 00248         BRB     26$                            ;
                            50      0000V    CF  9E 0024A 25$:    MOVAB   SCAN_DIRECT_1, R0              ;
                                             6E  DD 0024F 26$:    PUSHL   BUF_ADDRESS                    ; 5396
                                             52  DD 00251         PUSHL   PROC_LENGTH                    ;
                            60               02  FB 00253         CALLS   #2, (R0)                       ;
                                             0A  11 00256         BRB     28$                            ; 5392
                 00000000'  EF               01  D0 00258 27$:    MOVL    #1, DIRECTORY_ERROR            ; 5400
                                     0084    31     0025F         BRW     32$                            ; 5399
                            52 00000200      8F  C6 00262 28$:    DIVL2   #512, R2                       ; 5407
                            57               52  C0 00269         ADDL2   R2, READ_VBN                   ;
                            58               57  D1 0026C         CMPL    READ_VBN, R8                   ; 5408
                                             75  1A 0026F         BGTRU   32$                            ;
    0040   8F         00    6E               00  2C 00271         MOVC5   #0, (SP), #0, #64, FIB         ; 5413
                                     FE5C    CA     00278                                                ;
                      FE5C  CA 00200000      8F  D0 0027B         MOVL    #2097152, FIB                  ; 5414
                      FE60  CA         08    AA  D0 00284         MOVL    DIR_FID, FIB+4                 ; 5415
                      FE64  CA         0C    AA  B0 0028A         MOVW    DIR_FID+4, FIB+8               ; 5417
                                             7E  7C 00290         CLRQ    -(SP)                          ; 5422
                                             7E  7C 00292         CLRQ    -(SP)                          ;
                                             7E  D4 00294         CLRL    -(SP)                          ;
                                     C006    CF  9F 00296         PUSHAB  FIB_DESC                       ;
                                             7E  7C 0029A         CLRQ    -(SP)                          ;
                                       18    AA  9F 0029C         PUSHAB  IOSB                           ;
                            7E         72    8F  9A 0029F         MOVZBL  #114, -(SP)                    ;
                                00000000'    EF  DD 002A3         PUSHL   CHANNEL                        ;
                                             7E  D4 002A9         CLRL    -(SP)                          ;
                 00000000G  00               0C  FB 002AB         CALLS   #12, SYS$QIOW                  ;
                            56               50  D0 002B2         MOVL    R0, STATUS                     ;
                            0A               56  E9 002B5         BLBC    STATUS, 29$                    ; 5423
                            56         18    AA  3C 002B8         MOVZWL  IOSB, STATUS                   ;
                            03               56  E9 002BC         BLBC    STATUS, 29$                    ; 5424
                                     FEF7    31     002BF         BRW     19$                            ;
                 00000000'  EF               01  D0 002C2 29$:    MOVL    #1, DIRECTORY_ERROR            ; 5431
                                             56  DD 002C9         PUSHL   STATUS                         ; 5436
                                             6A  D5 002CB         TSTL    DIR_DESC                       ; 5435
                                             07  12 002CD         BNEQ    30$                            ;
                            50       BFA5    CF  9E 002CF         MOVAB   MFD_DESC, R0                   ;
                                             03  11 002D4         BRB     31$                            ;
                            50               6A  9E 002D6 30$:    MOVAB   DIR_DESC, R0                   ;
                                             50  DD 002D9 31$:    PUSHL   R0                             ;
                                             01  DD 002DB         PUSHL   #1                             ; 5432
                                00000000G    8F  DD 002DD         PUSHL   #VERIFY$_OPENDIR               ;
                            6B               04  FB 002E3         CALLS   #4, LIB$SIGNAL                 ;
                                             5E  DD 002E6 32$:    PUSHL   SP                             ; 5444
                                       08    AE  9F 002E8         PUSHAB  BUF_LENGTH                     ;
                 00000000G  00               02  FB 002EB         CALLS   #2, LIB$FREE_VM                ;
                            56               50  D0 002F2         MOVL    R0, STATUS                     ;
                            29               56  E8 002F5         BLBS    STATUS, 34$                    ; 5445
                                             56  DD 002F8         PUSHL   STATUS                         ;
```

```
                                    7E  D4 002FA          CLRL    -(SP)                                   ;
                         00000000G  8F  DD 002FC          PUSHL   #VERIFY$_FREEMEM                        ;
                                6B  03  FB 00302          CALLS   #3, LIB$SIGNAL                          ;
                                    1A  11 00305          BRB     34$                                     ; 5329
                                    7E  7C 00307 33$:     CLRQ    -(SP)                                   ; 5454
                                    7E  7C 00309          CLRQ    -(SP)                                   ;
                                    7E  7C 0030B          CLRQ    -(SP)                                   ;
                                    7E  7C 0030D          CLRQ    -(SP)                                   ;
                                7E      34  7D 0030F          MOVQ    #52, -(SP)                          ;
                         00000000'      EF  DD 00312          PUSHL   CHANNEL                             ;
                                    7E  D4 00318          CLRL    -(SP)                                   ;
                00000000G       00      0C  FB 0031A          CALLS   #12, SYS$QIOW                       ;
                                6A      59  D0 00321 34$:     MOVL    SAVE_DIR_DESC, DIR_DESC             ; 5461
                     08         AA   08  AE  D0 00324          MOVL    SAVE_DIR_FID, DIR_FID               ; 5462
                     0C         AA   0C  AE  B0 00329          MOVW    SAVE_DIR_FID+4, DIR_FID+4           ; 5464
                                        04  00 0032E          RET                                         ; 5465

; Routine Size:  815 bytes,    Routine Base:  CODE + 3D90
```

```
;  5489     5466  1 ROUTINE SCAN_DIRECT_1(LENGTH,ADDRESS)=
;  5490     5467  1
;  5491     5468  1 !++
;  5492     5469  1 !
;  5493     5470  1 ! FUNCTIONAL DESCRIPTION:
;  5494     5471  1 !       This routine scans an ODS-1 directory.
;  5495     5472  1 !
;  5496     5473  1 ! INPUT PARAMETERS:
;  5497     5474  1 !       LENGTH          - Descriptor for memory into wh'ch the directory
;  5498     5475  1 !       ADDRESS         -     has been read.
;  5499     5476  1 !
;  5500     5477  1 ! IMPLICIT INPUTS:
;  5501     5478  1 !       NONE
;  5502     5479  1 !
;  5503     5480  1 ! OUTPUT PARAMETERS:
;  5504     5481  1 !       NONE
;  5505     5482  1 !
;  5506     5483  1 ! IMPLICIT OUTPUTS:
;  5507     5484  1 !       NONE
;  5508     5485  1 !
;  5509     5486  1 ! ROUTINE VALUE:
;  5510     5487  1 !       True, indicating success.  There can be no directory format errors
;  5511     5488  1 !       in an ODS-1 directory.
;  5512     5489  1 !
;  5513     5490  1 ! SIDE EFFECTS:
;  5514     5491  1 !       Directory scan completed.
;  5515     5492  1 !
;  5516     5493  1 !--
;  5517     5494  1
;  5518     5495  2 BEGIN
;  5519     5496  2
;  5520     5497  2 ! Loop over all directory entries.
;  5521     5498  2 !
;  5522     5499  2 INCRA REC FROM .ADDRESS TO .ADDRESS + .LENGTH - NMB$C_DIRENTRY BY NMB$C_DIRENTRY DO
;  5523     5500  3     BEGIN
;  5524     5501  3     MAP
;  5525     5502  3         REC:            REF BBLOCK;     ! Pointer to directory record
;  5526     5503  3
;  5527     5504  3
;  5528     5505  3     ! Zero file number indicates an empty entry.  If nonzero, process the entry.
;  5529     5506  3     !
;  5530     5507  3     IF .REC[NMB$W_FID_NUM] NEQ 0
;  5531     5508  3     THEN
;  5532     5509  4         BEGIN
;  5533     5510  4         LOCAL
;  5534     5511  4             P,                                  ! Temporary
;  5535     5512  4             FILE_NAME:  VECTOR[FI2$S_FILENAME,BYTE],    ! Buffer for converted filename
;  5536     5513  4             VERSION:    BBLOCK[DIR$C_VERSION];  ! Dummy ODS-2 version entry
;  5537     5514  4
;  5538     5515  4
;  5539     5516  4         ! Convert the ODS-1 format entry to suitable parameters for
;  5540     5517  4         ! PROCESS_FILE and PROCESS_SUBDIR -- an ASCIC name string and
;  5541     5518  4         ! an ODS-2-format version entry.
;  5542     5519  4         !
;  5543     5520  4         FILE_NAME[0] = MAKE_STRING(,REC, FILE_NAME[1]);
;  5544     5521  4         P = CH$FIND_CH(.FILE_NAME[0], FILE_NAME[1], %C';');
;  5545     5522  4         IF .P NEQ 0 THEN FILE_NAME[0] = .P - FILE_NAME[1];
```

```
;  5546          5523  4          VERSION[DIR$W_VERSION] = .REC[NMB$W_VERSION];
;  5547          5524  4          VERSION[DIR$W_FID_NUM] = .REC[NMB$W_FID_NUM];
;  5548          5525  4          VERSION[DIR$W_FID_SEQ] = .REC[NMB$W_FID_SEQ];
;  5549          5526  4          VERSION[DIR$W_FID_RVN] = 1;
;  5550          5527  4
;  5551          5528  4
;  5552          5529  4          ! Process the entry.  If PROCESS_FILE determines that the entry is
;  5553          5530  4          ! valid, and it is a directory, recursively scan the directory.
;  5554          5531  4          ! Clear the directory bit so that each directory is only done once.
;  5555          5532  4          !
;  5556          5533  4          IF PROCESS_FILE(FILE_NAME, VERSION)
;  5557          5534  4          THEN
;  5558          5535  4              IF TESTBITSC(BITVECTOR[.DIRMAP[0], .REC[NMB$W_FID_NUM]-1])
;  5559          5536  4              THEN
;  5560          5537  4                  PROCESS_SUBDIR(FILE_NAME, VERSION);
;  5561          5538  3          END;
;  5562          5539  2      END;
;  5563          5540  2
;  5564          5541  2
;  5565          5542  2  ! All done, return success.
;  5566          5543  2  !
;  5567          5544  2  TRUE
;  5568          5545  1  END;


                                000C 00000 SCAN_DIRECT_1:
                                           .WORD   Save R2,R3                          ;  5466
                          5E    1C  C2 00002         SUBL2   #28, SP
                    50    08    AC  C1 00005         ADDL3   LENGTH, ADDRESS, R0       ;  5499
                          53    F0  A0 9E 0000B      MOVAB   -16(R0), R3
                          52    08  AC D0 0000F      MOVL    ADDRESS, REC
                                64  11 00013         BRB     5$
                                62  B5 00015  1$:     TSTW    (REC)                     ;  5507
                                5D  13 00017         BEQL    4$
                                09  AE 9F 00019      PUSHAB  FILE_NAME+1               ;  5520
                                52  DD 0001C         PUSHL   REC
                    00000000G   EF     02 FB 0001E   CALLS   #2, MAKE_STRING
                          08    AE     50 90 00025   MOVB    R0, FILE_NAME
                          50    08  AE 9A 00029      MOVZBL  FILE_NAME, R0             ;  5521
              09  AE          50     3B 3A 0002D     LOCC    #59, R0, FILE_NAME+1
                                02  12 00032         BNEQ    2$
                                51  D4 00034         CLRL    R1
                                51  D5 00036  2$:     TSTL    P                         ;  5522
                                09  13 00038         BEQL    3$
                          50    09  AE 9E 0003A      MOVAB   FILE_NAME+1, R0
              08  AE          51     50 83 0003E     SUBB3   R0, P, FILE_NAME
                          6E    0E  A2 B0 00043  3$:  MOVW    14(REC), VERSION          ;  5523
                    02    AE        62 D0 00047     MOVL    (REC), VERSION+2          ;  5524
                    06    AE        01 B0 0004B     MOVW    #1, VERSION+6             ;  5526
                                5E  DD 0004F         PUSHL   SP                        ;  5533
                                0C  AE 9F 00051      PUSHAB  FILE_NAME
                          FA64  CF     02 FB 00054   CALLS   #2, PROCESS_FILE
                                1A     50 E9 00059   BLBC    R0, 4$
                          51 00000000'  FF D0 0005C   MOVL    @DIRMAP, R1               ;  5535
```

```
                        50              62  3C 00063        MOVZWL  (REC), R0                       ;
                                        50  D7 00066        DECL    R0                              ;
                    0A          61      50  E5 00068        BBCC    R0, (R1), 4$                    ;
                                        5E  DD 0006C        PUSHL   SP                              ; 5537
                                0C      AE  9F 0006E        PUSHAB  FILE_NAME                       ;
                FC5B    CF              02  FB 00071        CALLS   #2, PROCESS_SUBDIR              ;
                        52              10  C0 00076 4$:    ADDL2   #16, REC                        ; 5499
                        53              52  D1 00079 5$:    CMPL    REC, R3                         ;
                                        97  1B 0007C        BLEQU   1$                              ;
                        50              01  D0 0007E        MOVL    #1, R0                          ; 5545
                                            04 00081        RET                                     ;

; Routine Size:  130 bytes,    Routine Base:  CODE + 40BF
```

```
;  5570     5546  1 ROUTINE SCAN_DIRECT_2(LENGTH,ADDRESS)=
;  5571     5547  1
;  5572     5548  1 !++
;  5573     5549  1 !
;  5574     5550  1 ! FUNCTIONAL DESCRIPTION:
;  5575     5551  1 !       This routine scans an ODS-2 directory.
;  5576     5552  1 !
;  5577     5553  1 ! INPUT PARAMETERS:
;  5578     5554  1 !       LENGTH          - Descriptor for memory into which the directory
;  5579     5555  1 !       ADDRESS         -     has been read.
;  5580     5556  1 !
;  5581     5557  1 ! IMPLICIT INPUTS:
;  5582     5558  1 !       NONE
;  5583     5559  1 !
;  5584     5560  1 ! OUTPUT PARAMETERS:
;  5585     5561  1 !       NONE
;  5586     5562  1 !
;  5587     5563  1 ! IMPLICIT OUTPUTS:
;  5588     5564  1 !       NONE
;  5589     5565  1 !
;  5590     5566  1 ! ROUTINE VALUE:
;  5591     5567  1 !       True if the directory was successfully scanned, false if an error
;  5592     5568  1 !       in its format is detected.
;  5593     5569  1 !
;  5594     5570  1 ! SIDE EFFECTS:
;  5595     5571  1 !       Directory scan completed.
;  5596     5572  1 !
;  5597     5573  1 !--
;  5598     5574  1
;  5599     5575  2 BEGIN
;  5600     5576  2 LOCAL
;  5601     5577  2         REC:            REF BBLOCK,     ! Pointer to directory record
;  5602     5578  2         NEXT_BLOCK;                     ! Pointer to next directory block
;  5603     5579  2
;  5604     5580  2
;  5605     5581  2 ! Initialize for the first block.
;  5606     5582  2 !
;  5607     5583  2 REC = .ADDRESS;
;  5608     5584  2 NEXT_BLOCK = .REC + 512;
;  5609     5585  2
;  5610     5586  2
;  5611     5587  2 ! Loop over all blocks.
;  5612     5588  2 !
;  5613     5589  2 WHILE .REC LSSA .ADDRESS + .LENGTH DO
;  5614     5590  3     BEGIN
;  5615     5591  3     IF .REC[DIR$W_SIZE] EQL 65535
;  5616     5592  3     THEN
;  5617     5593  4         BEGIN
;  5618     5594  4         !
;  5619     5595  4         ! End of this block.  Advance to next and resume.
;  5620     5596  4         !
;  5621     5597  4         REC = .NEXT_BLOCK;
;  5622     5598  4         NEXT_BLOCK = .REC + 512;
;  5623     5599  4         END
;  5624     5600  3     ELSE
;  5625     5601  4         BEGIN
;  5626     5602  4         LOCAL
```

```
;  5627     5603  4              NEXT_RECORD,                    ! Pointer to next record
;  5628     5604  4              VER:        REF BBLOCK;         ! Pointer to version entry
;  5629     5605  4
;  5630     5606  4
;  5631     5607  4          ! Point to where next record should start.  Make some validity tests
;  5632     5608  4          ! on the entry we are looking at.
;  5633     5609  4          !
;  5634     5610  4          NEXT_RECORD = .REC[DIR$W_SIZE] + .REC + 2;
;  5635     5611  4          IF
;  5636     5612  5              BEGIN
;  5637     5613  5              IF
;  5638     5614  5                  .NEXT_RECORD GEQA .NEXT_BLOCK OR                    ! Entry within block?
;  5639     5615  5                  .REC[DIR$W_SIZE] OR                                 ! Length even?
;  5640     5616  5                  .REC[DIR$W_SIZE] LSSU DIR$C_LENGTH + DIR$C_VERSION  ! Long enough?
;  5641     5617  5              THEN
;  5642     5618  5                  TRUE
;  5643     5619  5              ELSE
;  5644     5620  6                  BEGIN
;  5645     5621  6                  VER = (.REC + DIR$C_LENGTH + .REC[DIR$B_NAMECOUNT] + 1) AND NOT 1;
;  5646     5622  6                  .REC[DIR$V_TYPE] NEQ DIR$C_FID OR                   ! Proper type code?
;  5647     5623  6                  .VER GEQA .NEXT_BLOCK - DIR$C_VERSION               ! Version entry within block?
;  5648     5624  6                  END
;  5649     5625  5              END
;  5650     5626  4          THEN
;  5651     5627  5              BEGIN
;  5652     5628  5              !
;  5653     5629  5              ! Directory format error.  Report it and quit.
;  5654     5630  5              !
;  5655     5631  5              DIRECTORY_ERROR = TRUE;
;  5656     5632  5              SIGNAL(
;  5657     5633  5                  VERIFY$_BADDIR,
;  5658     5634  5                  1,
;  5659     5635  5                  (IF .DIR_DESC[0] EQL 0 THEN MFD_DESC ELSE DIR_DESC));
;  5660     5636  5              RETURN FALSE;
;  5661     5637  4              END;
;  5662     5638  4
;  5663     5639  4
;  5664     5640  4          ! Loop over all version entries.
;  5665     5641  4          !
;  5666     5642  4          WHILE .VER LSSA .NEXT_RECORD DO
;  5667     5643  5              BEGIN
;  5668     5644  5              LOCAL
;  5669     5645  5                  FILE_NUMBER,    ! File number of entry
;  5670     5646  5                  RVN;            ! RVN of entry
;  5671     5647  5
;  5672     5648  5
;  5673     5649  5              ! Get a clean file number and RVN.
;  5674     5650  5              !
;  5675     5651  5              FILE_NUMBER = .VER[DIR$W_FID_NUM];
;  5676     5652  5              FILE_NUMBER<16,8> = .VER[DIR$B_FID_NMX];
;  5677     5653  5              RVN = .VER[DIR$B_FID_RVN];
;  5678     5654  5              IF .RVN EQL 0 THEN RVN = .DIR_FID[FID$B_RVN];
;  5679     5655  5
;  5680     5656  5
;  5681     5657  5              ! Process the entry.  If PROCESS_FILE determines that the entry is
;  5682     5658  5              ! valid, and it is a directory, recursively scan the directory.
;  5683     5659  5              ! Clear the directory bit so that each directory is only done once.
```

```
;  5684          5660  5            !
;  5685          5661  5            IF PROCESS_FILE(REC[DIR$B_NAMECOUNT], VER[DIR$W_VERSION])
;  5686          5662  5            THEN
;  5687          5663  5                IF TESTBITSC(BITVECTOR[.DIRMAP[.RVN-1], .FILE_NUMBER-1])
;  5688          5664  5                THEN
;  5689          5665  6                    BEGIN
;  5690          5666  6                    IF
;  5691          5667  6                        CH$FIND_SUB(
;  5692          5668  6                            .REC[DIR$B_NAMECOUNT], REC[DIR$T_NAME],
;  5693          5669  6                            4, UPLIT BYTE ('.DIR')) EQL 0 OR
;  5694          5670  6                        .VER[DIR$W_VERSION] NEQ 1
;  5695          5671  6                    THEN
;  5696          5672  6                        SIGNAL(
;  5697          5673  6                            VERIFY$_DIRNAME,
;  5698          5674  6                            3,
;  5699          5675  6                            (IF .DIR_DESC[0] EQL 0 THEN MFD_DESC ELSE DIR_DESC),
;  5700          5676  6                            REC[DIR$B_NAMECOUNT],
;  5701          5677  6                            .VER[DIR$W_VERSION]);
;  5702          5678  6                    PROCESS_SUBDIR(REC[DIR$B_NAMECOUNT], VER[DIR$W_VERSION]);
;  5703          5679  5                    END;
;  5704          5680  5
;  5705          5681  5
;  5706          5682  5                ! Advance to next version entry.
;  5707          5683  5                !
;  5708          5684  5                VER = .VER + DIR$C_VERSION;
;  5709          5685  4                END;
;  5710          5686  4
;  5711          5687  4
;  5712          5688  4            ! Advance to next directory record.
;  5713          5689  4            !
;  5714          5690  4            REC = .NEXT_RECORD;
;  5715          5691  3            END;
;  5716          5692  2        END;
;  5717          5693  2
;  5718          5694  2
;  5719          5695  2    ! All done, return success.
;  5720          5696  2    !
;  5721          5697  2    TRUE
;  5722          5698  1    END;


                                52  49  44  2E  04141 P.ABR:  .ASCII  \.DIR\                                    ;


                                                    0FFC 00000 SCAN_DIRECT_2:
                                                               .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11        ; 5546
                                    5B 00000000G    00  9E 00002           MOVAB   LIB$SIGNAL, R11                ;
                                    5A      BEB6    CF  9E 00009           MOVAB   MFD_DESC, R10                  ;
                                    59 00000000'    EF  9E 0000E           MOVAB   DIR_DESC, R9                   ;
                                    54        08    AC  D0 00015           MOVL    ADDRESS, REC                   ; 5583
                                    55      0200    C4  9E 00019           MOVAB   512(R4), NEXT_BLOCK            ; 5584
                              58          08  AC      04  AC  C1 0001E       ADDL3   LENGTH, ADDRESS, R8            ; 5589
                                    58              54  D1 00024 1$:        CMPL    REC, R8                        ;
                                                    03  1F 00027           BLSSU   2$                             ;
                                                00E7    31 00029           BRW     16$                            ;
```

```
                        50           64  3C 0002C 28:     MOVZWL  (REC), R0                       ; 5591
                    FFFF 8F          50  B1 0002F         CMPW    R0, #65535
                                     0A  12 00034         BNEQ    3$
                        54           55  D0 00036         MOVL    NEXT_BLOCK, REC                 ; 5597
                        55     0200  C4  9E 00039         MOVAB   512(R4), NEXT_BLOCK             ; 5598
                                     E4  11 0003E         BRB     1$                              ; 5591
                        57       02 A440 9E 00040 38:     MOVAB   2(REC)[R0], NEXT_RECORD         ; 5610
                        55           57  D1 00045         CMPL    NEXT_RECORD, NEXT_BLOCK         ; 5614
                                     24  1E 00048         BGEQU   4$
                        21           50  E8 0004A         BLBS    R0, 4$                          ; 5615
                        0E           50  B1 0004D         CMPW    R0, #14                         ; 5616
                                     1C  1F 00050         BLSSU   4$
                        50       05  A4  9A 00052         MOVZBL  5(REC), R0                      ; 5621
                        50       07 A044 9E 00056         MOVAB   7(R0)[REC], R0
                    56  50           01  CB 0005B         BICL3   #1, R0, VER
                        07       04  A4  93 0005F         BITB    4(REC), #7                      ; 5622
                                     09  12 00063         BNEQ    4$
                        50       F8  A5  9E 00065         MOVAB   -8(R5), R0                      ; 5623
                        50           56  D1 00069         CMPL    VER, R0
                                     23  1F 0006C         BLSSU   7$
                00000000'    EF          01  D0 0006E 48:     MOVL    #1, DIRECTORY_ERROR         ; 5631
                                         69  D5 00075         TSTL    DIR_DESC                    ; 5635
                                         05  12 00077         BNEQ    5$
                        50               6A  9E 00079         MOVAB   MFD_DESC, R0
                                         03  11 0007C         BRB     6$
                        50               69  9E 0007E 58:     MOVAB   DIR_DESC, R0
                                         50  DD 00081 68:     PUSHL   R0
                                         01  DD 00083         PUSHL   #1                          ; 5632
                            00000000G    8F  DD 00085         PUSHL   #VERIFY$_BADDIR
                        6B               03  FB 00088         CALLS   #3, LIB$SIGNAL
                                         50  D4 0008E         CLRL    R0                          ; 5636
                                             04 00090         RET
                        57               56  D1 00091 78:     CMPL    VER, NEXT_RECORD            ; 5642
                                         77  1E 00094         BGEQU   15$
                        52       02      A6  3C 00096         MOVZWL  2(VER), FILE_NUMBER         ; 5651
                52      10       07      A6  F0 0009A         INSV    7(VER), #16, #8, FILE_NUMBER ; 5652
                        53       06      A6  9A 000A0         MOVZBL  6(VER), RVN                 ; 5653
                                         04  12 000A4         BNEQ    8$                          ; 5654
                        53       0C      A9  9A 000A6         MOVZBL  DIR_FID+4, RVN
                                         56  DD 000AA 88:     PUSHL   VER                         ; 5661
                                 05      A4  9F 000AC         PUSHAB  5(REC)
                            F983         CF  02 FB 000AF         CALLS   #2, PROCESS_FILE
                        51               50  E9 000B4         BLBC    R0, 14$
                        50       0084 D943   DE 000B7         MOVAL   @DIRMAP[RVN], R0            ; 5663
                                         52  D7 000BD         DECL    R2
                    44      FC           B0  52 E5 000BF         BBCC    R2, @-4(R0), 14$
                        50               05  A4 9A 000C4         MOVZBL  5(REC), R0              ; 5668
                06  A4  50       4139    CA  04 39 000C8         MATCHC  #4, P.ABR, R0, 6(REC)   ; 5669
                                         03  13 000D0         BEQL    9$
                        53               04  D0 000D2         MOVL    #4, R3
                        53               04  C2 000D5 98:     SUBL2   #4, R3
                                         05  13 000D8         BEQL    10$
                        01               66  B1 000DA         CMPW    (VER), #1                   ; 5670
                                         1F  13 000DD         BEQL    13$
                        7E               66  32 000DF 108:    CVTWL   (VER), -(SP)                ; 5677
                                 05      A4  9F 000E2         PUSHAB  5(REC)                      ; 5676
                                         69  D5 000E5         TSTL    DIR_DESC                    ; 5675
```

```
                                 05  12  000E7          BNEQ    11$                                     ;
                              50 6A  9E  000E9          MOVAB   MFD_DESC, R0                            ;
                                 03  11  000EC          BRB     12$                                     ;
                              50 69  9E  000EE  11$:     MOVAB   DIR_DESC, R0                           ;
                                 50  DD  000F1  12$:     PUSHL   R0                                      ;
                                 03  DD  000F3          PUSHL   #3                                      ; 5676
                        00000000G 8F  DD  000F5          PUSHL   #VERIFY$_DIRNAME                        ;
                              6B    05  FB  000FB          CALLS   #5, LIB$SIGNAL                          ;
                                 56  DD  000FE  13$:     PUSHL   VER                                     ; 5678
                              05    A4  9F  00100          PUSHAB  5(REC)                                  ;
                    .     FB43  CF    02  FB  00103          CALLS   #2, PROCESS_SUBDIR                      ;
                              56    08  C0  00108  14$:     ADDL2   #8, VER                                 ; 5684
                                 84  11  0010B          BRB     7$                                      ; 5642
                              54 57  D0  0010D  15$:     MOVL    NEXT_RECORD, REC                        ; 5690
                                 FF11 31  00110          BRW     1$                                      ; 5589
                              50 01  D0  00113  16$:     MOVL    #1, R0                                  ; 5698
                                 04  00116          RET                                             ;

; Routine Size:  279 bytes,    Routine Base:  CODE + 4145
```

```
;  5724          5699  1 ROUTINE DIR_SCAN(RVN): NOVALUE=
;  5725          5700  1
;  5726          5701  1 !++
;  5727          5702  1 !
;  5728          5703  1 ! FUNCTIONAL DESCRIPTION:
;  5729          5704  1 !       This routine scans all directories on a volume.
;  5730          5705  1 !
;  5731          5706  1 ! INPUT PARAMETERS:
;  5732          5707  1 !       RVN             - Relative volume number.
;  5733          5708  1 !
;  5734          5709  1 ! IMPLICIT INPUTS:
;  5735          5710  1 !       NONE
;  5736          5711  1 !
;  5737          5712  1 ! OUTPUT PARAMETERS:
;  5738          5713  1 !       NONE
;  5739          5714  1 !
;  5740          5715  1 ! IMPLICIT OUTPUTS:
;  5741          5716  1 !       NONE
;  5742          5717  1 !
;  5743          5718  1 ! ROUTINE VALUE:
;  5744          5719  1 !       NONE
;  5745          5720  1 !
;  5746          5721  1 ! SIDE EFFECTS:
;  5747          5722  1 !       Directory scan completed.
;  5748          5723  1 !
;  5749          5724  1 !--
;  5750          5725  1
;  5751          5726  2 BEGIN
;  5752          5727  2 LOCAL
;  5753          5728  2         VERSION:        BBLOCK[DIR$C_VERSION];  ! Dummy version entry
;  5754          5729  2
;  5755          5730  2
;  5756          5731  2 ! Initialize descriptor for current directory string.
;  5757          5732  2 !
;  5758          5733  2 DIR_DESC[0] = 0;
;  5759          5734  2 DIR_DESC[1] = DIR;
;  5760          5735  2
;  5761          5736  2
;  5762          5737  2 ! Initialize current directory ID.
;  5763          5738  2 !
;  5764          5739  2 DIR_FID[FID$W_NUM] = FID$C_MFD;
;  5765          5740  2 DIR_FID[FID$W_SEQ] = FID$C_MFD;
;  5766          5741  2 DIR_FID[FID$W_RVN] = .RVN;
;  5767          5742  2
;  5768          5743  2
;  5769          5744  2 ! Initialize dummy version entry pointing to MFD.
;  5770          5745  2 !
;  5771          5746  2 VERSION[DIR$W_VERSION] = 1;
;  5772          5747  2 VERSION[DIR$W_FID_NUM] = FID$C_MFD;
;  5773          5748  2 VERSION[DIR$W_FID_SEQ] = FID$C_MFD;
;  5774          5749  2 VERSION[DIR$W_FID_RVN] = .RVN;
;  5775          5750  2
;  5776          5751  2
;  5777          5752  2 ! Process the MFD.
;  5778          5753  2 !
;  5779          5754  2 PROCESS_SUBDIR(0, VERSION);
;  5780          5755  1 END;
```

```
                                0004 00000 DIR_SCAN:
                                                        .WORD   Save R2                                 ; 5699
                     52 00000000'   EF  9E 00002        MOVAB   DIR_DESC, R2
                     5E             04  C2 00009        SUBL2   #4, SP
                                    62  D4 0000C        CLRL    DIR_DESC                                ; 5733
                  04 A2      FEC0   C2  9E 0000E        MOVAB   DIR, DIR_DESC+4                         ; 5734
                  08 A2 00040004    8F  DD 00014        MOVL    #262148, DIR_FID                        ; 5739
                  0C A2        04   AC  B0 0001C        MOVW    RVN, DIR_FID+4                          ; 5741
                       00040001    8F  DD 00021        PUSHL   #262145                                 ; 5746
                  04 AE            04  B0 00027        MOVW    #4, VERSION+4                           ; 5748
                  06 AE        04   AC  B0 0002B        MOVW    RVN, VERSION+6                          ; 5749
                                    5E  DD 00030        PUSHL   SP                                      ; 5754
                                    7E  D4 00032        CLRL    -(SP)
                FAFB CF             02  FB 00034        CALLS   #2, PROCESS_SUBDIR
                                        04 00039        RET                                             ; 5755

; Routine Size:  58 bytes,    Routine Base:  CODE + 425C
```

```
;  5782     5756  1 ROUTINE FAO(CTRL,PARAM): NOVALUE=
;  5783     5757  1
;  5784     5758  1 !++
;  5785     5759  1 !
;  5786     5760  1 ! FUNCTIONAL DESCRIPTION:
;  5787     5761  1 !       This routine interfaces to FAO to format information into the line.
;  5788     5762  1 !
;  5789     5763  1 ! INPUT PARAMETERS:
;  5790     5764  1 !       CTRL            - ASCIC control string
;  5791     5765  1 !       PARAM...        - Parameters required by the control string (if any)
;  5792     5766  1 !
;  5793     5767  1 ! IMPLICIT INPUTS:
;  5794     5768  1 !       LIST_DESC       - Describes the remainder of the output line.
;  5795     5769  1 !
;  5796     5770  1 ! OUTPUT PARAMETERS:
;  5797     5771  1 !       NONE
;  5798     5772  1 !
;  5799     5773  1 ! IMPLICIT OUTPUTS:
;  5800     5774  1 !       LIST_DESC       - Updated.
;  5801     5775  1 !       Information formatted into the line buffer.
;  5802     5776  1 !
;  5803     5777  1 ! ROUTINE VALUE:
;  5804     5778  1 !       NONE
;  5805     5779  1 !
;  5806     5780  1 ! SIDE EFFECTS:
;  5807     5781  1 !       NONE
;  5808     5782  1 !
;  5809     5783  1 !--
;  5810     5784  1
;  5811     5785  2 BEGIN
;  5812     5786  2 MAP
;  5813     5787  2         CTRL:           REF VECTOR[,BYTE];      ! ASCIC control string
;  5814     5788  2 LOCAL
;  5815     5789  2         OUTLEN:         WORD,                   ! Length returned by $FAOL
;  5816     5790  2         DESC:           VECTOR[2];              ! Descriptor for control string
;  5817     5791  2
;  5818     5792  2
;  5819     5793  2 ! Make a descriptor for the control string.
;  5820     5794  2 !
;  5821     5795  2 DESC[0] = .CTRL[0];
;  5822     5796  2 DESC[1] = CTRL[1];
;  5823     5797  2
;  5824     5798  2
;  5825     5799  2 ! Use $FAOL to do the editing.
;  5826     5800  2 !
;  5827     5801  2 $FAOL(CTRSTR=DESC, OUTLEN=OUTLEN, OUTBUF=LIST_DESC, PRMLST=PARAM);
;  5828     5802  2
;  5829     5803  2
;  5830     5804  2 ! Update the listing buffer descriptor.
;  5831     5805  2 !
;  5832     5806  2 LIST_DESC[0] = .LIST_DESC[0] - .OUTLEN;
;  5833     5807  2 LIST_DESC[1] = .LIST_DESC[1] + .OUTLEN;
;  5834     5808  1 END;


                                                        .EXTRN  SYS$FAOL
```

```
                                                0004 00000 FAO:     .WORD   Save R2                                  ; 5756
                                 52 00000000'   EF   9E 00002       MOVAB   LIST_DESC, R2                           ;
                                 5E             0C   C2 00009       SUBL2   #12, SP                                 ;
                           04    AE       04    BC   9A 0000C       MOVZBL  @CTRL, DESC                             ; 5795
                     08    AE    04    AC          01   C1 00011    ADDL3   #1, CTRL, DESC+4                        ; 5796
                                             08    AC   9F 00017    PUSHAB  PARAM                                   ; 5801
                                             52    DD   9F 0001A    PUSHL   R2                                      ;
                                             08    AE   9F 0001C    PUSHAB  OUTLEN                                  ;
                                             10    AE   9F 0001F    PUSHAB  DESC                                    ;
                           00000000G         00          04   FB 00022    CALLS   #4, SYS$FAOL                        ;
                                             50          6E   3C 00029    MOVZWL  OUTLEN, R0                          ; 5806
                                             62          50   C2 0002C    SUBL2   R0, LIST_DESC                       ;
                                             50          6E   3C 0002F    MOVZWL  OUTLEN, R0                          ; 5807
                                       04    A2          50   C0 00032    ADDL2   R0, LIST_DESC+4                     ;
                                                         04   00036       RET                                         ; 5808

; Routine Size:  55 bytes,    Routine Base:  CODE + 4296
```

```
;  5836          5809  1 ROUTINE EOL: NOVALUE=
;  5837          5810  1
;  5838          5811  1 !++
;  5839          5812  1 !
;  5840          5813  1 ! FUNCTIONAL DESCRIPTION:
;  5841          5814  1 !       This routine writes the listing buffer to the listing file.
;  5842          5815  1 !
;  5843          5816  1 ! INPUT PARAMETERS:
;  5844          5817  1 !       NONE
;  5845          5818  1 !
;  5846          5819  1 ! IMPLICIT INPUTS:
;  5847          5820  1 !       LIST_DESC       - Describes the remainder of the listing line.
;  5848          5821  1 !
;  5849          5822  1 ! OUTPUT PARAMETERS:
;  5850          5823  1 !       NONE
;  5851          5824  1 !
;  5852          5825  1 ! IMPLICIT OUTPUTS:
;  5853          5826  1 !       LIST_DESC       - Reinitialized to describe the entire line.
;  5854          5827  1 !
;  5855          5828  1 ! ROUTINE VALUE:
;  5856          5829  1 !       NONE
;  5857          5830  1 !
;  5858          5831  1 ! SIDE EFFECTS:
;  5859          5832  1 !       The listing is produced.
;  5860          5833  1 !
;  5861          5834  1 !--
;  5862          5835  1
;  5863          5836  2 BEGIN
;  5864          5837  2
;  5865          5838  2 ! Compute line length.
;  5866          5839  2 !
;  5867          5840  2 LIST_RAB[RAB$W_RSZ] = LIST_SIZE - .LIST_DESC[0];
;  5868          5841  2
;  5869          5842  2
;  5870          5843  2 ! Do the write.  If failure, report it.
;  5871          5844  2 !
;  5872          5845  2 IF NOT $PUT(RAB=LIST_RAB)
;  5873          5846  2 THEN
;  5874          5847  2     FILE_ERROR(
;  5875          5848  2         VERIFY$_FACILITY^16 + SHR$_WRITEERR + STS$K_SEVERE,
;  5876          5849  2         LIST_FAB,
;  5877          5850  2         .LIST_RAB[RAB$L_STS], .LIST_RAB[RAB$L_STV]);
;  5878          5851  2
;  5879          5852  2
;  5880          5853  2 ! Reinitialize descriptor.
;  5881          5854  2 !
;  5882          5855  2 LIST_DESC[0] = LIST_SIZE;
;  5883          5856  2 LIST_DESC[1] = LIST_BUFFER;
;  5884          5857  1 END;


                                  0004 00000 EOL:     .WORD   Save R2                                 ; 5809
                    52 00000000'  EF  9E 00002        MOVAB   LIST_DESC, R2                           ;
              FE7E  C2      0084  8F      62  A3 00009        SUBW3   LIST_DESC, #132, LIST_RAB+34            ; 5840
```

```
                          FE5C   C2  9F 00011          PUSHAB  LIST_RAB                                                  ; 5845
                00000000G 00     01  FB 00015          CALLS   #1, SYS$PUT                                               ;
                          14     50  E8 0001C          BLBS    R0, 1$                                                    ;
                          7E FE64   C2  7D 0001F          MOVQ    LIST_RAB+8, -(SP)                                         ; 5850
                             FE0C   C2  9F 00024          PUSHAB  LIST_FAB                                                  ; 5847
                      00000000*     8F  DD 00028          PUSHL   #<<<VERIFY$_FACILITY@16>+4304>+4>                         ; 5848
                        F6B3  CF     04  FB 0002E          CALLS   #4, FILE_ERROR                                            ;
                          62     84  8F  9A 00033 1$:      MOVZBL  #132, LIST_DESC                                           ; 5855
                          04  A2     08  A2  9E 00037          MOVAB   LIST_BUFFER, LIST_DESC+4                              ; 5856
                                         04  0003C          RET                                                               ; 5857

; Routine Size:  61 bytes,    Routine Base:  CODE + 42CD
```

```
;  5886     5858  1 ROUTINE ENTER_WORK(TYPE,P1,P2): NOVALUE=
;  5887     5859  1
;  5888     5860  1 !++
;  5889     5861  1 !
;  5890     5862  1 ! FUNCTIONAL DESCRIPTION:
;  5891     5863  1 !       This routine makes an entry in the work list that is executed after
;  5892     5864  1 !       the volume is unlocked.
;  5893     5865  1 !
;  5894     5866  1 ! INPUT PARAMETERS:
;  5895     5867  1 !       TYPE            - Type of repair.
;  5896     5868  1 !       P1              - Type-specific parameters.
;  5897     5869  1 !       P2              -
;  5898     5870  1 !
;  5899     5871  1 ! IMPLICIT INPUTS:
;  5900     5872  1 !       WORK_LIST       - List header for work items.
;  5901     5873  1 !
;  5902     5874  1 ! OUTPUT PARAMETERS:
;  5903     5875  1 !       NONE
;  5904     5876  1 !
;  5905     5877  1 ! IMPLICIT OUTPUTS:
;  5906     5878  1 !       WORK_LIST       - List header for work items updated.
;  5907     5879  1 !
;  5908     5880  1 ! ROUTINE VALUE:
;  5909     5881  1 !       NONE
;  5910     5882  1 !
;  5911     5883  1 ! SIDE EFFECTS:
;  5912     5884  1 !       NONE
;  5913     5885  1 !
;  5914     5886  1 !--
;  5915     5887  1
;  5916     5888  2 BEGIN
;  5917     5889  2 MAP
;  5918     5890  2         P1:             REF BBLOCK,     ! Type-specific parameters
;  5919     5891  2         P2:             REF BBLOCK;     !
;  5920     5892  2 LOCAL
;  5921     5893  2         P:              REF BBLOCK,     ! Pointer to work list entry
;  5922     5894  2         STATUS;                         ! Status variable
;  5923     5895  2 BIND
;  5924     5896  2         SIZES = UPLIT BYTE (            ! Table of entry sizes
;  5925     5897  2             WRK_S_ENTER,
;  5926     5898  2             WRK_S_REMOVE,
;  5927     5899  2             WRK_S_ADDQUO,
;  5928     5900  2             WRK_S_DELETE)
;  5929     5901  2         : VECTOR[,BYTE];
;  5930     5902  2
;  5931     5903  2
;  5932     5904  2 ! Allocate a new block.
;  5933     5905  2 !
;  5934     5906  2 STATUS = LIB$GET_VM(%REF(.SIZES[.TYPE]), P);
;  5935     5907  2 IF NOT .STATUS THEN SIGNAL(VERIFY$_ALLOCMEM, 0, .STATUS);
;  5936     5908  2
;  5937     5909  2
;  5938     5910  2 ! Link the block to the work list.  Offset 0 points to the first entry and
;  5939     5911  2 ! offset 1 points to the most recent entry.
;  5940     5912  2 !
;  5941     5913  2 IF .WORK_LIST[0] EQL 0
;  5942     5914  2     THEN WORK_LIST[0] = .P
```

```
;  5943          5915  2      ELSE BBLOCK[.WORK_LIST[1], WRK_LINK] = .P;
;  5944          5916  2 WORK_LIST[1] = .P;
;  5945          5917  2
;  5946          5918  2
;  5947          5919  2 ! Initialize the remainder.
;  5948          5920  2 !
;  5949          5921  2 P[WRK_LINK] = 0;
;  5950          5922  2 P[WRK_TYPE] = .TYPE;
;  5951          5923  2 CASE .TYPE FROM WRK_K_ENTER TO WRK_K_DELETE OF
;  5952          5924  2     SET
;  5953          5925  2
;  5954          5926  2     [WRK_K_ENTER, WRK_K_DELETE]:
;  5955          5927  2         BEGIN
;  5956          5928  2         BBLOCK[P[WRK_FID], FID$W_NUM] = .P1[FID$W_NUM];
;  5957          5929  2         BBLOCK[P[WRK_FID], FID$W_SEQ] = .P1[FID$W_SEQ];
;  5958          5930  2         BBLOCK[P[WRK_FID], FID$W_RVN] = .P1[FID$W_RVN];
;  5959          5931  2         END;
;  5960          5932  2
;  5961          5933  2     [WRK_K_REMOVE]:
;  5962          5934  2         BEGIN
;  5963          5935  2         BBLOCK[P[WRK_DID], FID$W_NUM] = .P2[FID$W_NUM];
;  5964          5936  2         BBLOCK[P[WRK_DID], FID$W_SEQ] = .P2[FID$W_SEQ];
;  5965          5937  2         BBLOCK[P[WRK_DID], FID$W_RVN] = .P2[FID$W_RVN];
;  5966          5938  2         BBLOCK[P[WRK_FID], FID$W_NUM] = .P1[FID$W_NUM];
;  5967          5939  2         BBLOCK[P[WRK_FID], FID$W_SEQ] = .P1[FID$W_SEQ];
;  5968          5940  2         BBLOCK[P[WRK_FID], FID$W_RVN] = .P1[FID$W_RVN];
;  5969          5941  2         END;
;  5970          5942  2
;  5971          5943  2     [WRK_K_ADDQUO]:
;  5972          5944  2         BEGIN
;  5973          5945  2         P[WRK_UIC] = .P1;
;  5974          5946  2         P[WRK_USAGE] = .P2;
;  5975          5947  2         END;
;  5976          5948  2
;  5977          5949  2     TES;
;  5978          5950  1 END;

                                0C  10  12  0C  0430A P.ABS:  .BYTE   12, 18, 16, 12                          ;
                                                      SIZES=          P.ABS

                                                001C 00000 ENTER_WORK:
                                                           .WORD   Save R2,R3,R4                          ; 5858
                                    54 00000000'   EF   9E 00002           MOVAB   WORK_LIST, R4
                                    5E             08   C2 00009           SUBL2   #8, SP
                                           04      AE   9F 0000C           PUSHAB  P                      ; 5906
                                    50             EA   AF 0000F           MOVAB   SIZES, R0
                                04  AE             04 BC40 9A 00013        MOVZBL  @TYPE[R0], 4(SP)
                                           04      AE   9F 00019           PUSHAB  4(SP)
                            00000000G  00             02   FB 0001C        CALLS   #2, LIB$GET_VM
                                    11                50   E8 00023        BLBS    STATUS, 1$             ; 5907
                                                      50   DD 00026        PUSHL   STATUS
                                                      7E   D4 00028        CLRL    -(SP)
                                       00000000G      8F   DD 0002A        PUSHL   #VERIFY$_ALLOCMEM
```

```
                    00000000G  00          03 FB 00030          CALLS   #3, LIB$SIGNAL                          ;
                               50       04 AE D0 00037 1$:      MOVL    P, R0                                  ; 5914
                                  64       D5 0003B             TSTL    WORK_LIST                              ; 5913
                                  05       12 0003D             BNEQ    2$                                     ;
                               64          50 D0 0003F          MOVL    R0, WORK_LIST                          ; 5914
                                  04       11 00042             BRB     3$                                     ;
                            04 B4          50 D0 00044 2$:      MOVL    R0, @WORK_LIST+4                       ; 5915
                            04 A4          50 D0 00048 3$:      MOVL    R0, WORK_LIST+4                        ; 5916
                                  60       D4 0004C             CLRL    (R0)                                   ; 5921
                            04 A0       04 AC 90 0004E          MOVB    TYPE, 4(R0)                            ; 5922
                               53       08 AC D0 00053          MOVL    P1, R3                                 ; 5928
                      03       00       04 AC CF 00057          CASEL   TYPE, #0, #3                           ; 5923
          0008        0031        0015        0008             0005C 4$:  .WORD   5$-4$,-
                                                                           6$-4$,-
                                                                           7$-4$,-
                                                                           5$-4$
                               51       06 A0 9E 00064 5$:      MOVAB   6(R0), R1                              ; 5928
                               61          63 D0 00068          MOVL    (R3), (R1)                             ;
                            04 A1       04 A3 B0 0006B          MOVW    4(R3), 4(R1)                           ; 5930
                                  04       04 00070             RET                                            ; 5923
                               52       0C A0 9E 00071 6$:      MOVAB   12(R0), R2                             ; 5935
                               51       0C AC D0 00075          MOVL    P2, R1                                 ;
                               62          61 D0 00079          MOVL    (R1), (R2)                             ;
                            04 A2       04 A1 B0 0007C          MOVW    4(R1), 4(R2)                           ; 5937
                               50          06 C0 00081          ADDL2   #6, R0                                 ; 5938
                               60          63 D0 00084          MOVL    (R3), (R0)                             ;
                            04 A0       04 A3 B0 00087          MOVW    4(R3), 4(R0)                           ; 5940
                                  04       04 0008C             RET                                            ; 5923
                            08 A0          53 D0 0008D 7$:      MOVL    R3, 8(R0)                              ; 5945
                            0C A0       0C AC D0 00091          MOVL    P2, 12(R0)                             ; 5946
                                  04       04 00096             RET                                            ; 5950

; Routine Size:  151 bytes,    Routine Base:  CODE + 430E
```

```
;  5980          5951  1 ROUTINE PROCESS_WORK: NOVALUE=
;  5981          5952  1
;  5982          5953  1 !++
;  5983          5954  1 !
;  5984          5955  1 ! FUNCTIONAL DESCRIPTION:
;  5985          5956  1 !       This routine processes the work list to execute delayed repairs.
;  5986          5957  1 !
;  5987          5958  1 ! INPUT PARAMETERS:
;  5988          5959  1 !       NONE
;  5989          5960  1 !
;  5990          5961  1 ! IMPLICIT INPUTS:
;  5991          5962  1 !       WORK_LIST       - List header for work items.
;  5992          5963  1 !
;  5993          5964  1 ! OUTPUT PARAMETERS:
;  5994          5965  1 !       NONE
;  5995          5966  1 !
;  5996          5967  1 ! IMPLICIT OUTPUTS:
;  5997          5968  1 !       NONE
;  5998          5969  1 !
;  5999          5970  1 ! ROUTINE VALUE:
;  6000          5971  1 !       NONE
;  6001          5972  1 !
;  6002          5973  1 ! SIDE EFFECTS:
;  6003          5974  1 !       NONE
;  6004          5975  1 !
;  6005          5976  1 !--
;  6006          5977  1
;  6007          5978  2 BEGIN
;  6008          5979  2 LOCAL
;  6009          5980  2         STATUS,                         ! Status variable
;  6010          5981  2         P:              REF BBLOCK;     ! Pointer to work list entry
;  6011          5982  2 LABEL
;  6012          5983  2         ENTER_LOST;
;  6013          5984  2
;  6014          5985  2
;  6015          5986  2 P = .WORK_LIST[0];
;  6016          5987  2 WHILE .P NEQ 0 DO
;  6017          5988  3     BEGIN
;  6018          5989  3     CASE .P[WRK_TYPE] FROM WRK_K_ENTER TO WRK_K_DELETE OF
;  6019          5990  3         SET
;  6020          5991  3
;  6021          5992  3
;  6022          5993  3         [WRK_K_ENTER]:
;  6023          5994  3 ENTER_LOST:
;  6024          5995  4                 BEGIN
;  6025          5996  4                 LOCAL
;  6026          5997  4                     IDENT_AREA:     REF BBLOCK,     ! Pointer to ident area
;  6027          5998  4                     FILENAME:       VECTOR[FI2$S_FILENAME + FI2$S_FILENAMEXT + 1, BYTE],
;  6028          5999  4                                                     ! Buffer for file name
;  6029          6000  4                     FNA_DESC:       VECTOR[2];      ! Descriptor for file name
;  6030          6001  4
;  6031          6002  4
;  6032          6003  4                 ! Reread the file header to get a file name.
;  6033          6004  4                 !
;  6034          6005  4                 IF READ_HEADER(P[WRK_FID], BUFFER_2)
;  6035          6006  4                 THEN
;  6036          6007  5                     BEGIN
```

```
;  6037     6008  5
;  6038     6009  5                    ! If this is the first pass through, locate the lost file
;  6039     6010  5                    ! directory, creating it if necessary.
;  6040     6011  5                    !
;  6041     6012  5                    IF .LOST_DIR_FID[FID$W_NUM] EQL 0
;  6042     6013  5                    THEN
;  6043     6014  6                        BEGIN
;  6044     6015  6                        OWN
;  6045     6016  6                            ATR_UIC:        BBLOCK[4],
;  6046     6017  6                            ATR_FPRO:       BBLOCK[2],
;  6047     6018  6                            ATR_UCHAR:      BBLOCK[4],
;  6048     6019  6                            ATR_RECATTR:    BBLOCK[FAT$C_LENGTH];
;  6049     6020  6                        BIND
;  6050     6021  6                            ATR_DESC = UPLIT(
;  6051     6022  6                                WORD(4, ATR$C_UIC), LONG(ATR_UIC),
;  6052     6023  6                                WORD(2, ATR$C_FPRO), LONG(ATR_FPRO),
;  6053     6024  6                                WORD(4, ATR$C_UCHAR), LONG(ATR_UCHAR),
;  6054     6025  6                                WORD(FAT$C_LENGTH, ATR$C_RECATTR), LONG(ATR_RECATTR),
;  6055     6026  6                                LONG(0));
;  6056     6027  6
;  6057     6028  6
;  6058     6029  6                        ! Access the MFD on RVN 1 to get attributes.  The lost file
;  6059     6030  6                        ! directory will be created with the same attributes.
;  6060     6031  6                        !
;  6061     6032  6                        CH$FILL(0, FIB$C_LENGTH, FIB);
;  6062     6033  6                        FIB[FIB$W_FID_NUM] = FID$C_MFD;
;  6063     6034  6                        FIB[FIB$W_FID_SEQ] = FID$C_MFD;
;  6064     6035  6                        FIB[FIB$W_FID_RVN] = 1;
;  6065   P 6036  6                        STATUS = $QIOW(
;  6066   P 6037  6                            FUNC=IO$_ACCESS,
;  6067   P 6038  6                            CHAN=.CHANNEL,
;  6068   P 6039  6                            IOSB=IOSB,
;  6069   P 6040  6                            P1=FIB_DESC,
;  6070     6041  6                            P5=ATR_DESC);
;  6071     6042  6                        IF .STATUS THEN STATUS = .IOSB[0];
;  6072     6043  6                        IF NOT .STATUS
;  6073     6044  6                        THEN
;  6074     6045  7                            BEGIN
;  6075     6046  7                            SIGNAL(VERIFY$_CREATELOST, 0, .STATUS);
;  6076     6047  7                            LEAVE ENTER_LOST;
;  6077     6048  6                            END;
;  6078     6049  6
;  6079     6050  6
;  6080     6051  6                        ! Adjust EFBLK and HIBLK.
;  6081     6052  6                        !
;  6082     6053  6                        ATR_RECATTR[FAT$L_EFBLK] = 0;
;  6083     6054  6                        ATR_RECATTR[FAT$L_HIBLK] = 0;
;  6084     6055  6                        IF .STRUCTURE_LEVEL EQL 2 THEN ATR_RECATTR[FAT$L_EFBLK] = 2 ^ 16;
;  6085     6056  6
;  6086     6057  6
;  6087     6058  6                        ! Access the directory file, creating it if necessary.
;  6088     6059  6                        !
;  6089     6060  6                        CH$FILL(0, FIB$C_LENGTH, FIB);
;  6090     6061  6                        FIB[FIB$L_ACCTL] = FIB$M_WRITE OR FIB$M_NOWRITE;
;  6091     6062  6                        FIB[FIB$W_DID_NUM] = FID$C_MFD;
;  6092     6063  6                        FIB[FIB$W_DID_SEQ] = FID$C_MFD;
;  6093     6064  6                        FIB[FIB$W_DID_RVN] = 1;
```

```
;  6094      6065  6                    IF .STRUCTURE_LEVEL EQL 2
;  6095      6066  6                    THEN
;  6096      6067  7                        BEGIN
;  6097      6068  7                        FIB[FIB$W_EXCTL] = FIB$M_EXTEND OR FIB$M_ALCON OR FIB$M_FILCON;
;  6098      6069  7                        FIB[FIB$L_EXSZ] = 1;
;  6099      6070  6                        END;
;  6100    P 6071  6                    STATUS = $QIOW(
;  6101    P 6072  6                        FUNC=IO$_ACCESS OR IO$M_CREATE OR IO$M_ACCESS,
;  6102    P 6073  6                        CHAN=.CHANNEL,
;  6103    P 6074  6                        IOSB=IOSB,
;  6104    P 6075  6                        P1=FIB_DESC,
;  6105    P 6076  6                        P2=LOST_DESC,
;  6106      6077  6                        P5=ATR_DESC);
;  6107      6078  6                    IF .STATUS THEN STATUS = .IOSB[0];
;  6108      6079  6                    IF NOT .STATUS
;  6109      6080  6                    THEN
;  6110      6081  7                        BEGIN
;  6111      6082  7                        SIGNAL(VERIFY$_CREATELOST, 0, .STATUS);
;  6112      6083  7                        LEAVE ENTER_LOST;
;  6113      6084  6                        END;
;  6114      6085  6
;  6115      6086  6
;  6116      6087  6                    ! If the directory file was created and it is ODS-2, the
;  6117      6088  6                    ! first block must be initialized.
;  6118      6089  6                    !
;  6119      6090  6                    IF .STATUS EQL SS$_CREATED THEN IF .STRUCTURE_LEVEL EQL 2
;  6120      6091  6                    THEN
;  6121      6092  7                        BEGIN
;  6122    P 6093  7                        STATUS = $QIOW(
;  6123    P 6094  7                            FUNC=IO$_WRITEVBLK,
;  6124    P 6095  7                            CHAN=.CHANNEL,
;  6125    P 6096  7                            IOSB=IOSB,
;  6126    P 6097  7                            P1=UPLIT WORD(-1, REP 255 OF (0)),
;  6127    P 6098  7                            P2=512,
;  6128      6099  7                            P3=1);
;  6129      6100  7                        IF .STATUS THEN STATUS = .IOSB[0];
;  6130      6101  7                        IF NOT .STATUS
;  6131      6102  7                        THEN
;  6132      6103  8                            BEGIN
;  6133      6104  8                            SIGNAL(VERIFY$_CREATELOST, 0, .STATUS);
;  6134      6105  8                            LEAVE ENTER_LOST;
;  6135      6106  7                            END;
;  6136      6107  6                        END;
;  6137      6108  6
;  6138      6109  6
;  6139      6110  6                    ! Save the file ID for later use.
;  6140      6111  6                    !
;  6141      6112  6                    LOST_DIR_FID[FID$W_NUM] = .FIB[FIB$W_FID_NUM];
;  6142      6113  6                    LOST_DIR_FID[FID$W_SEQ] = .FIB[FIB$W_FID_SEQ];
;  6143      6114  6                    LOST_DIR_FID[FID$W_RVN] = .FIB[FIB$W_FID_RVN];
;  6144      6115  6
;  6145      6116  6
;  6146      6117  6                    ! Deaccess the lost file directory.
;  6147      6118  6                    !
;  6148    P 6119  6                    $QIOW(
;  6149    P 6120  6                        FUNC=IO$_DEACCESS,
;  6150      6121  6                        CHAN=.CHANNEL);
```

```
;  6151          6122  5                    END;
;  6152          6123  5
;  6153          6124  5
;  6154          6125  5                    ! Get a descriptor for the file name.
;  6155          6126  5                    !
;  6156          6127  5                    IDENT_AREA = BUFFER_2 + .BUFFER_2[FH2$B_IDOFFSET]*2;
;  6157          6128  5                    IF .STRUCTURE_LEVEL EQL 2
;  6158          6129  5                    THEN
;  6159          6130  6                        BEGIN
;  6160          6131  6                        CH$COPY(
;  6161          6132  6                            FI2$S_FILENAME, IDENT_AREA[FI2$T_FILENAME],
;  6162          6133  6                            %C' ',
;  6163          6134  6                            FI2$S_FILENAME + FI2$S_FILENAMEXT + 1, FILENAME);
;  6164          6135  6
;  6165          6136  6                        IF (.BUFFER_2[FH2$B_MPOFFSET] - .BUFFER_2[FH2$B_IDOFFSET]) * 2
;  6166          6137  6                        GEQU $BYTEOFFSET(FI2$T_FILENAMEXT) + FI2$S_FILENAMEXT
;  6167          6138  6                        THEN
;  6168          6139  6                            CH$MOVE(
;  6169          6140  6                                FI2$S_FILENAMEXT,
;  6170          6141  6                                IDENT_AREA[FI2$T_FILENAMEXT],
;  6171          6142  6                                FILENAME[FI2$S_FILENAME]);
;  6172          6143  6
;  6173          6144  6                        FNA_DESC[0] =
;  6174          6145  6                            CH$FIND_CH(FI2$S_FILENAME + FI2$S_FILENAMEXT + 1, FILENAME, %C' ')
;  6175          6146  6                            - FILENAME;
;  6176          6147  6                        END
;  6177          6148  5                    ELSE
;  6178          6149  6                        BEGIN
;  6179          6150  6                        FNA_DESC[0] = MAKE_STRING(
;  6180          6151  6                            IDENT_AREA[FI1$W_FILENAME] - $BYTEOFFSET(NMB$W_NAME),
;  6181          6152  6                            FILENAME);
;  6182          6153  5                        END;
;  6183          6154  5                    FNA_DESC[1] = FILENAME;
;  6184          6155  5
;  6185          6156  5
;  6186          6157  5                    ! Create the directory entry.
;  6187          6158  5                    !
;  6188          6159  5                    CH$FILL(0, FIB$C_LENGTH, FIB);
;  6189          6160  5                    FIB[FIB$W_FID_NUM] = .BBLOCK[P[WRK_FID], FID$W_NUM];
;  6190          6161  5                    FIB[FIB$W_FID_SEQ] = .BBLOCK[P[WRK_FID], FID$W_SEQ];
;  6191          6162  5                    FIB[FIB$W_FID_RVN] = .BBLOCK[P[WRK_FID], FID$W_RVN];
;  6192          6163  5                    FIB[FIB$W_DID_NUM] = .LOST_DIR_FID[FID$W_NUM];
;  6193          6164  5                    FIB[FIB$W_DID_SEQ] = .LOST_DIR_FID[FID$W_SEQ];
;  6194          6165  5                    FIB[FIB$W_DID_RVN] = .LOST_DIR_FID[FID$W_RVN];
;  6195          6166  5                    FIB[FIB$W_NMCTL] = FIB$M_NEWVER;
;  6196        P 6167  5                    STATUS = $QIOW(
;  6197        P 6168  5                        FUNC=IO$_CREATE,
;  6198        P 6169  5                        CHAN=.CHANNEL,
;  6199        P 6170  5                        IOSB=IOSB,
;  6200        P 6171  5                        P1=FIB_DESC,
;  6201          6172  5                        P2=FNA_DESC);
;  6202          6173  5                    IF .STATUS THEN STATUS = .IOSB[0];
;  6203          6174  5                    IF NOT .STATUS
;  6204          6175  5                    THEN
;  6205          6176  5                        HEADER_ERROR(VERIFY$_ENTERLOST, P[WRK_FID], 0, .STATUS);
;  6206          6177  4                    END;
;  6207          6178  3                END;
```

```
;  6208          6179  3
;  6209          6180  3
;  6210          6181  3          [WRK_K_REMOVE]:
;  6211          6182  4              BEGIN
;  6212          6183  4
;  6213          6184  4              ! Remove directory entry.
;  6214          6185  4              !
;  6215          6186  4              CH$FILL(0, FIB$C_LENGTH, FIB);
;  6216          6187  4              FIB[FIB$W_FID_NUM] = .BBLOCK[P[WRK_FID], FID$W_NUM];
;  6217          6188  4              FIB[FIB$W_FID_SEQ] = .BBLOCK[P[WRK_FID], FID$W_SEQ];
;  6218          6189  4              FIB[FIB$W_FID_RVN] = .BBLOCK[P[WRK_FID], FID$W_RVN];
;  6219          6190  4              FIB[FIB$W_DID_NUM] = .BBLOCK[P[WRK_DID], FID$W_NUM];
;  6220          6191  4              FIB[FIB$W_DID_SEQ] = .BBLOCK[P[WRK_DID], FID$W_SEQ];
;  6221          6192  4              FIB[FIB$W_DID_RVN] = .BBLOCK[P[WRK_DID], FID$W_RVN];
;  6222          6193  4              FIB[FIB$W_NMCTL] = FIB$M_FINDFID;
;  6223        P 6194  4              STATUS = $QIOW(
;  6224        P 6195  4                  FUNC=IO$_DELETE,
;  6225        P 6196  4                  CHAN=.CHANNEL,
;  6226        P 6197  4                  IOSB=IOSB,
;  6227          6198  4                  P1=FIB_DESC);
;  6228          6199  4              IF .STATUS THEN STATUS = .IOSB[0];
;  6229          6200  4              IF NOT .STATUS
;  6230          6201  4              THEN
;  6231          6202  4                  SIGNAL(
;  6232          6203  4                      VERIFY$_REMOVE,
;  6233          6204  4                      3,
;  6234          6205  4                      .BBLOCK[P[WRK_FID], FID$W_NUM] +
;  6235          6206  4                      .BBLOCK[P[WRK_FID], FID$B_NMX] ^ 16,
;  6236          6207  4                      .BBLOCK[P[WRK_FID], FID$W_SEQ],
;  6237          6208  4                      .BBLOCK[P[WRK_FID], FID$B_RVN],
;  6238          6209  4                      .STATUS);
;  6239          6210  3              END;
;  6240          6211  3
;  6241          6212  3
;  6242          6213  3          [WRK_K_ADDQUO]:
;  6243          6214  4              BEGIN
;  6244          6215  4
;  6245          6216  4              ! Add quota file entry.
;  6246          6217  4              !
;  6247          6218  4              CH$FILL(0, FIB$C_LENGTH, FIB);
;  6248          6219  4              FIB[FIB$W_CNTRLFUNC] = FIB$C_ADD_QUOTA;
;  6249          6220  4              DQF[DQF$L_UIC] = .P[WRK_UIC];
;  6250          6221  4              DQF[DQF$L_USAGE] = .P[WRK_USAGE];
;  6251          6222  4              DQF[DQF$L_PERMQUOTA] = .DEFAULT_QUOTA;
;  6252          6223  4              DQF[DQF$L_OVERDRAFT] = .DEFAULT_OVERDRAFT;
;  6253        P 6224  4              STATUS = $QIOW(
;  6254        P 6225  4                  FUNC=IO$_ACPCONTROL,
;  6255        P 6226  4                  CHAN=.CHANNEL,
;  6256        P 6227  4                  IOSB=IOSB,
;  6257        P 6228  4                  P1=FIB_DESC,
;  6258          6229  4                  P2=DQF_DESC);
;  6259          6230  4              IF .STATUS THEN STATUS = .IOSB[0];
;  6260          6231  4              IF NOT .STATUS
;  6261          6232  4              THEN
;  6262          6233  4                  SIGNAL(
;  6263          6234  4                      VERIFY$_ADDQUOTA,
;  6264          6235  4                      1,
```

```
;  6265          6236  4                       .P[WRK_UIC],
;  6266          6237  4                       .STATUS);
;  6267          6238  3                   END;
;  6268          6239  3
;  6269          6240  3
;  6270          6241  3               [WRK_K_DELETE]:
;  6271          6242  4                   BEGIN
;  6272          6243  4
;  6273          6244  4                   ! Delete file.
;  6274          6245  4                   !
;  6275          6246  4                   CH$FILL(0, FIB$C_LENGTH, FIB);
;  6276          6247  4                   FIB[FIB$W_FID_NUM] = .BBLOCK[P[WRK_FID], FID$W_NUM];
;  6277          6248  4                   FIB[FIB$W_FID_SEQ] = .BBLOCK[P[WRK_FID], FID$W_SEQ];
;  6278          6249  4                   FIB[FIB$W_FID_RVN] = .BBLOCK[P[WRK_FID], FID$W_RVN];
;  6279        P 6250  4                   STATUS = $QIOW(
;  6280        P 6251  4                       FUNC=IO$_DELETE OR IO$M_DELETE,
;  6281        P 6252  4                       CHAN=.CHANNEL,
;  6282        P 6253  4                       IOSB=IOSB,
;  6283          6254  4                       P1=FIB_DESC);
;  6284          6255  4                   IF .STATUS THEN STATUS = .IOSB[0];
;  6285          6256  4                   IF NOT .STATUS
;  6286          6257  4                   THEN
;  6287          6258  4                       HEADER_ERROR(VERIFY$_DELETE, P[WRK_FID], 0, .STATUS);
;  6288          6259  3                   END;
;  6289          6260
;  6290          6261
;  6291          6262               TES;
;  6292          6263
;  6293          6264
;  6294          6265           ! Advance to next work list entry.
;  6295          6266           !
;  6296          6267           P = .P[WRK_LINK];
;  6297          6268       END;
;  6298          6269   1 END;


                                            .PSECT  DATA,NOEXE,2

                                08B2F               .BLKB   1
                                08B30 ATR_UIC:      .BLKB   4
                                08B34 ATR_FPRO:
                                                    .BLKB   2
                                08B36               .BLKB   2
                                08B38 ATR_UCHAR:
                                                    .BLKB   4
                                08B3C ATR_RECATTR:
                                                    .BLKB   32

                                                    .PSECT  CODE,NOWRT,2

                                043A5               .BLKB   3
                0015  0004      043A8 P.ABT:        .WORD   4, 21
                  00000000'     043AC               .ADDRESS ATR_UIC
                0016  0002      043B0               .WORD   2, 22
                  00000000'     043B4               .ADDRESS ATR_FPRO
                0003  0004      043B8               .WORD   4, 3
```

```
                                     00000000' 043BC            .ADDRESS ATR_UCHAR                                 ;
                                0004  0020  043C0            .WORD    32, 4                                        ;
                                     00000000' 043C4            .ADDRESS ATR_RECATTR                               ;
                                     00000000  043C8            .LONG    0                                         ;
                                          FFFF  043CC P.ABU:    .WORD    -1                                        ;
                                         0000#  043CE           .WORD    0[255]                                    ;

                                                    ATR_DESC=            P.ABT


                                          OFFC 00000 PROCESS_WORK:
                                                             .WORD    Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11       ; 5951
                       5B 00000000'   EF   9E 00002          MOVAB    IOSB, R11                                  ;
                       5E         A0  AE   9E 00009          MOVAB    -96(SP), SP                                ;
                       58         14  AB   D0 0000D          MOVL     WORK_LIST, P                               ; 5986
                                      01   12 00011 1$:      BNEQ     2$                                         ; 5987
                                           04 00013          RET                                                 ;
                 03    00         04  A8   8F 00014 2$:      CASEB    4(P), #0, #3                               ; 5989
    0306         02A1          021E      0008    00019 3$:   .WORD    4$-3$,-                                    ;
                                                                      18$-3$,-                                   ;
                                                                      21$-3$,-                                   ;
                                                                      25$-3$                                     ;
                             FC44  CB   9F 00021 4$:         PUSHAB   BUFFER_2                                   ; 6005
                       5A         06  A8   9E 00025          MOVAB    6(P), R10                                  ;
                                      5A   DD 00029          PUSHL    R10                                        ;
                 F02A  CF             02   FB 0002B          CALLS    #2, READ_HEADER                            ;
                       03             50   E8 00030          BLBS     R0, 5$                                     ;
                                    0341   31 00033          BRW      28$                                        ;
                               F8     AB   B5 00036 5$:      TSTW     LOST_DIR_FID                               ; 6012
                                      03   13 00039          BEQL     6$                                         ;
                                    0136   31 0003B          BRW      12$                                        ;
    0040   8F          00      6E     00   2C 0003E 6$:      MOVC5    #0, (SP), #0, #64, FIB                     ; 6032
                             FE44     CB      00045                                                              ;
                 FE48  CB 00040004    8F   D0 00048          MOVL     #262148, FIB+4                             ; 6033
                 FE4C  CB             01   B0 00051          MOVW     #1, FIB+8                                  ; 6035
                                      7E   D4 00056          CLRL     -(SP)                                      ; 6041
                             FD80     CF   9F 00058          PUSHAB   ATR_DESC                                   ;
                                      7E   7C 0005C          CLRQ     -(SP)                                      ;
                                      7E   D4 0005E          CLRL     -(SP)                                      ;
                             BA00     CF   9F 00060          PUSHAB   FIB_DESC                                   ;
                                      7E   7C 00064          CLRQ     -(SP)                                      ;
                                      5B   DD 00066          PUSHL    R11                                        ;
                                      32   DD 00068          PUSHL    #50                                        ;
                         00000000'    EF   DD 0006A          PUSHL    CHANNEL                                    ;
                                      7E   D4 00070          CLRL     -(SP)                                      ;
            00000000G  00             0C   FB 00072          CALLS    #12, SYS$QIOW                              ;
                       59             50   D0 00079          MOVL     R0, STATUS                                 ;
                       7A             59   E9 0007C          BLBC     STATUS, 9$                                 ; 6042
                       59             6B   3C 0007F          MOVZWL   IOSB, STATUS                               ;
                       74             59   E9 00082          BLBC     STATUS, 9$                                 ; 6043
                             00D8     CB   7C 00085          CLRQ     ATR_RECATTR+4                              ; 6054
                                      56   D4 00089          CLRL     R6                                         ; 6055
                       02         0C  AB   D1 0008B          CMPL     STRUCTURE_LEVEL, #2                        ;
                                      0B   12 0008F          BNEQ     7$                                         ;
                                      56   D6 00091          INCL     R6                                         ;
                 00DC  CB 00020000    8F   D0 00093          MOVL     #131072, ATR_RECATTR+8                     ;
    0040   8F          00      6E     00   2C 0009C 7$:      MOVC5    #0, (SP), #0, #64, FIB                     ; 6060
```

```
                        FE44    CB      000A3
                FE44    CB      0101    8F  3C  000A6       MOVZWL  #257, FIB                                  ;  6061
                FE4E    CB  00040004    8F  D0  000AD       MOVL    #262148, FIB+10                             ;  6062
                FE52    CB              01  B0  000B6       MOVW    #1, FIB+14                                  ;  6064
                        0B              56  E9  000BB       BLBC    R6, 8$                                      ;  6065
                FE5A    CB        85    8F  9B  000BE       MOVZBW  #133, FIB+22                                ;  6068
                FE5C    CB              01  D0  000C4       MOVL    #1, FIB+24                                  ;  6069
                                        7E  D4  000C9 8$:   CLRL    -(SP)                                       ;  6077
                              FDDD      CF  9F  000CB       PUSHAB  ATR_DESC
                                        7E  7C  000CF       CLRQ    -(SP)
                              B99F      CF  9F  000D1       PUSHAB  LOST_DESC
                              B98B      CF  9F  000D5       PUSHAB  FIB_DESC
                                        7E  7C  000D9       CLRQ    -(SP)
                                        5B  DD  000DB       PUSHL   R11
                        7E        F2    8F  9A  000DD       MOVZBL  #242, -(SP)
                            00000000'   EF  DD  000E1       PUSHL   CHANNEL
                                        7E  D4  000E7       CLRL    -(SP)
            00000000G   00              0C  FB  000E9       CALLS   #12, SYS$QIOW
                        59              50  D0  000F0       MOVL    R0, STATUS
                        44              59  E9  000F3       BLBC    STATUS, 10$                                 ;  6078
                        59              6B  3C  000F6       MOVZWL  IOSB, STATUS
                        3E              59  E9  000F9 9$:   BLBC    STATUS, 10$                                 ;  6079
            00000619    8F              59  D1  000FC       CMPL    STATUS, #1561                               ;  6090
                                        49  12  00103       BNEQ    11$
                        02        0C    AB  D1  00105       CMPL    STRUCTURE_LEVEL, #2
                                        43  12  00109       BNEQ    11$
                                        7E  7C  0010B       CLRQ    -(SP)                                       ;  6099
                        7E              01  7D  0010D       MOVQ    #1, -(SP)
                        7E        0200  8F  3C  00110       MOVZWL  #512, -(SP)
                              FCE7      CF  9F  00115       PUSHAB  P.ABU
                                        7E  7C  00119       CLRQ    -(SP)
                                        5B  DD  0011B       PUSHL   R11
                                        30  DD  0011D       PUSHL   #48
                            00000000'   EF  DD  0011F       PUSHL   CHANNEL
                                        7E  D4  00125       CLRL    -(SP)
            00000000G   00              0C  FB  00127       CALLS   #12, SYS$QIOW
                        59              50  D0  0012E       MOVL    R0, STATUS
                        06              59  E9  00131       BLBC    STATUS, 10$                                 ;  6100
                        59              6B  3C  00134       MOVZWL  IOSB, STATUS
                        14              59  E8  00137       BLBS    STATUS, 11$                                 ;  6101
                                        59  DD  0013A 10$:   PUSHL   STATUS                                     ;  6104
                                        7E  D4  0013C       CLRL    -(SP)
                            00000000G   8F  DD  0013E       PUSHL   #VERIFY$_CREATELOST
            00000000G   00              03  FB  00144       CALLS   #3, LIB$SIGNAL
                              0229      31  31  00148       BRW     28$                                         ;  6105
                F8  AB      FE48    CB  D0  00014E 11$:     MOVL    FIB+4, LOST_DIR_FID                         ;  6112
                FC  AB      FE4C    CB  B0  00154       MOVW    FIB+8, LOST_DIR_FID+4                       ;  6114
                                        7E  7C  0015A       CLRQ    -(SP)                                       ;  6121
                                        7E  7C  0015C       CLRQ    -(SP)
                                        7E  7C  0015E       CLRQ    -(SP)
                                        7E  7C  00160       CLRQ    -(SP)
                        7E              34  7D  00162       MOVQ    #52, -(SP)
                            00000000'   EF  DD  00165       PUSHL   CHANNEL
                                        7E  D4  0016B       CLRL    -(SP)
            00000000G   00              0C  FB  0016D       CALLS   #12, SYS$QIOW
                        56      FC44    CB  9A  00174 12$:  MOVZBL  BUFFER_2, R6                                ;  6127
                        57      FC44 CB46   3E  00179       MOVAW   BUFFER_2[R6], IDENT_AREA
```

```
                                                02      0C     AB  D1 0017F          CMPL    STRUCTURE_LEVEL, #2                          ;6128
                                                                  3A  12 00183          BNEQ    15$
0057    8F              20                      67                14  2C 00185          MOVC5   #20, (IDENT_AREA), #32, #87, FILENAME ;6132
                                                        08        AE      0018C
                                                50      FC45      CB  9A 0018E          MOVZBL  BUFFER_2+1, R0                       ;6136
                                                50                56  C2 00193          SUBL2   R6, R0
                                                50                02  C4 00196          MULL2   #2, R0
                                  00000078      8F                50  D1 00199          CMPL    R0, #120                             ;6137
                                                                  08  1F 001A0          BLSSU   13$
                1C      AE              36      A7        0042    8F  28 001A2          MOVC3   #66, 54(IDENT_AREA), FILENAME+20     ;6142
                08      AE      0057    8F                        20  3A 001AA 13$:     LOCC    #32, #87, FILENAME                   ;6145
                                                                  02  12 001B1          BNEQ    14$
                                                                  51  D4 001B3          CLRL    R1
                                                50        08      AE  9E 001B5 14$:     MOVAB   FILENAME, R0                         ;6146
                        6E                      51                50  C3 001B9          SUBL3   R0, R1, FNA_DESC
                                                                  10  11 001BD          BRB     16$                                  ;6128
                                                          08      AE  9F 001BF 15$:     PUSHAB  FILENAME                             ;6150
                                                          FA      A7  9F 001C2          PUSHAB  -6(IDENT_AREA)                       ;6151
                                  00000000G     EF                02  FB 001C5          CALLS   #2, MAKE_STRING
                                                6E                50  D0 001CC          MOVL    R0, FNA_DESC
                                          04    AE        08      AE  9E 001CF 16$:     MOVAB   FILENAME, FNA_DESC+4                 ;6154
0040    8F              00                      6E                00  2C 001D4          MOVC5   #0, (SP), #0, #64, FIB               ;6159
                                                        FE44      CB      001DB
                                    FE48        CB                6A  D0 001DE          MOVL    (R10), FIB+4                         ;6160
                                    FE4C        CB        04      AA  B0 001E3          MOVW    4(R10), FIB+8                        ;6162
                                    FE4E        CB        F8      AB  D0 001E9          MOVL    LOST_DIR_FID, FIB+10                 ;6163
                                    FE52        CB        FC      AB  B0 001EF          MOVW    LOST_DIR_FID+4, FIB+14               ;6165
                                    FE58        CB      0200      8F  B0 001F5          MOVW    #512, FIB+20                         ;6166
                                                                  7E  7C 001FC          CLRQ    -(SP)                                ;6172
                                                                  7E  7C 001FE          CLRQ    -(SP)
                                                          10      AE  9F 00200          PUSHAB  FNA_DESC
                                                        B85D      CF  9F 00203          PUSHAB  FIB_DESC
                                                                  7E  7C 00207          CLRQ    -(SP)
                                                                  5B  DD 00209          PUSHL   R11
                                                                  33  DD 0020B          PUSHL   #51
                                                00000000'         EF  DD 0020D          PUSHL   CHANNEL
                                                                  7E  D4 00213          CLRL    -(SP)
                                  00000000G     00                0C  FB 00215          CALLS   #12, SYS$QIOW
                                                59                50  D0 0021C          MOVL    R0, STATUS
                                                06                59  E9 0021F          BLBC    STATUS, 17$                          ;6173
                                                59                6B  3C 00222          MOVZWL  IOSB, STATUS
                                                69                59  E8 00225          BLBS    STATUS, 19$                          ;6174
                                                                  59  DD 00228 17$:     PUSHL   STATUS                               ;6176
                                                                  7E  D4 0022A          CLRL    -(SP)
                                                                  5A  DD 0022C          PUSHL   R10
                                                00000000G         8F  DD 0022E          PUSHL   #VERIFY$_ENTERLOST
                                                          013B    31      00234         BRW     27$
0040    8F              00                      6E                00  2C 00237 18$:     MOVC5   #0, (SP), #0, #64, FIB               ;6186
                                                        FE44      CB      0023E
                                                52        06      A8  9E 00241          MOVAB   6(P), R2                             ;6187
                                                53                62  3C 00245          MOVZWL  (R2), R3
                                    FE48        CB                53  B0 00248          MOVW    R3, FIB+4
                                    FE4A        CB        02      A2  D0 0024D          MOVL    2(R2), FIB+6                         ;6188
                                                50        0C      A8  9E 00253          MOVAB   12(P), R0                            ;6190
                                    FE4E        CB                60  D0 00257          MOVL    (R0), FIB+10
                                    FE52        CB        04      A0  B0 0025C          MOVW    4(R0), FIB+14                        ;6192
                                    FE58        CB      0800      8F  B0 00262          MOVW    #2048, FIB+20                        ;6193
```

```
                                  7E  7C 00269          CLRQ    -(SP)                            ; 6198
                                  7E  7C 0026B          CLRQ    -(SP)                            ;
                                  7E  D4 0026D          CLRL    -(SP)                            ;
                          B7F1    CF  9F 0026F          PUSHAB  FIB_DESC                         ;
                                  7E  7C 00273          CLRQ    -(SP)                            ;
                                  5B  DD 00275          PUSHL   R11                              ;
                                  35  DD 00277          PUSHL   #53                              ;
                     00000000'    EF  DD 00279          PUSHL   CHANNEL                          ;
                                  7E  D4 0027F          CLRL    -(SP)                            ;
          00000000G  00           0C  FB 00281          CALLS   #12, SYS$QIOW                    ;
                     59           50  D0 00288          MOVL    R0, STATUS                       ;
                     06           59  E9 0028B          BLBC    STATUS, 20$                      ; 6199
                     59           6B  3C 0028E          MOVZWL  IOSB, STATUS                     ;
                     72           59  E8 00291 19$:     BLBS    STATUS, 22$                      ; 6200
                                  59  DD 00294 20$:     PUSHL   STATUS                           ; 6209
                     7E     04    A2  9A 00296          MOVZBL  4(R2), -(SP)                     ; 6208
                     7E     02    A2  3C 0029A          MOVZWL  2(R2), -(SP)                     ; 6207
                     52     05    A2  9A 0029E          MOVZBL  5(R2), R2                        ; 6206
            52       52           10  78 002A2          ASHL    #16, R2, R2                      ;
                               6243   9F 002A6          PUSHAB  (R2)[R3]                         ; 6205
                                  03  DD 002A9          PUSHL   #3                               ; 6202
                     00000000G    8F  DD 002AB          PUSHL   #VERIFY$_REMOVE                  ;
          00000000G  00           06  FB 002B1          CALLS   #6, LIB$SIGNAL                   ;
                                  63  11 002B8          BRB     24$                              ; 5989
 0040  8F           00  6E        00  2C 002BA 21$:     MOVC5   #0, (SP), #0, #64, FIB           ; 6218
                          FE44    CB     002C1                                                   ;
          FE5A  CB                0B  B0 002C4          MOVW    #11, FIB+22                      ; 6219
     00000000'  EF          08    A8  7D 002C9          MOVQ    8(P), DQF+4                      ; 6220
     00000000'  EF   00000000'    EF  7D 002D1          MOVQ    DEFAULT_QUOTA, DQF+12            ; 6222
                                  7E  7C 002DC          CLRQ    -(SP)                            ; 6229
                                  7E  7C 002DE          CLRQ    -(SP)                            ;
                          B788    CF  9F 002E0          PUSHAB  DQF_DESC                         ;
                          B77C    CF  9F 002E4          PUSHAB  FIB_DESC                         ;
                                  7E  7C 002E8          CLRQ    -(SP)                            ;
                                  5B  DD 002EA          PUSHL   R11                              ;
                                  38  DD 002EC          PUSHL   #56                              ;
                     00000000'    EF  DD 002EE          PUSHL   CHANNEL                          ;
                                  7E  D4 002F4          CLRL    -(SP)                            ;
          00000000G  00           0C  FB 002F6          CALLS   #12, SYS$QIOW                    ;
                     59           50  D0 002FD          MOVL    R0, STATUS                       ;
                     06           59  E9 00300          BLBC    STATUS, 23$                      ; 6230
                     59           6B  3C 00303          MOVZWL  IOSB, STATUS                     ;
                     6E           59  E8 00306 22$:     BLBS    STATUS, 28$                      ; 6231
                                  59  DD 00309 23$:     PUSHL   STATUS                           ; 6237
                            08    A8  DD 0030B          PUSHL   8(P)                             ; 6236
                                  01  DD 0030E          PUSHL   #1                               ; 6233
                     00000000G    8F  DD 00310          PUSHL   #VERIFY$_ADDQUOTA                ;
          00000000G  00           04  FB 00316          CALLS   #4, LIB$SIGNAL                   ;
                                  58  11 0031D 24$:     BRB     28$                              ; 5989
 0040  8F           00  6E        00  2C 0031F 25$:     MOVC5   #0, (SP), #0, #64, FIB           ; 6246
                          FE44    CB     00326                                                   ;
                     52     06    A8  9E 00329          MOVAB   6(P), R2                         ; 6247
          FE48  CB                62  D0 0032D          MOVL    (R2), FIB+4                      ;
          FE4C  CB          04    A2  B0 00332          MOVW    4(R2), FIB+8                     ; 6249
                                  7E  7C 00338          CLRQ    -(SP)                            ; 6254
                                  7E  7C 0033A          CLRQ    -(SP)                            ;
                                  7E  D4 0033C          CLRL    -(SP)                            ;
```

```
                                B722    CF  9F 0033E          PUSHAB  FIB_DESC                                ;
                                        7E  7C 00342          CLRQ    -(SP)                                   ;
                                        5B  DD 00344          PUSHL   R11                                     ;
                                7E   0135  8F  3C 00346          MOVZWL  #309, -(SP)                             ;
                                    00000000'   EF  DD 0034B          PUSHL   CHANNEL                         ;
                                        7E  D4 00351          CLRL    -(SP)                                   ;
                        00000000G   00      0C  FB 00353          CALLS   #12, SYS$QIOW                       ;
                                59          50  D0 0035A          MOVL    R0, STATUS                          ;
                                06          59  E9 0035D          BLBC    STATUS, 26$                         ; 6255
                                59          6B  3C 00360          MOVZWL  IOSB, STATUS                        ;
                                11          59  E8 00363          BLBS    STATUS, 28$                         ; 6256
                                        59  DD 00366 26$:       PUSHL   STATUS                              ; 6258
                                        7E  D4 00368          CLRL    -(SP)                                   ;
                                        52  DD 0036A          PUSHL   R2                                      ;
                                    00000000G   8F  DD 0036C          PUSHL   #VERIFY$_DELETE                 ;
                            FODD    CF          04  FB 00372 27$:       CALLS   #4, HEADER_ERROR                ;
                                58          68  D0 00377 28$:       MOVL    (P), P                              ; 6267
                                        FC94    31  0037A          BRW     1$                                  ; 5987
                                                04  0037D          RET                                         ; 6269

; Routine Size:  894 bytes,    Routine Base:  CODE + 45CC
```

```
;  6300     6270  1 ROUTINE DO_REPAIR(P_PROMPT)=
;  6301     6271  1
;  6302     6272  1 !++
;  6303     6273  1 !
;  6304     6274  1 ! FUNCTIONAL DESCRIPTION:
;  6305     6275  1 !       This routine evaluates the qualifiers that specify whether a repair
;  6306     6276  1 !       should be executed.
;  6307     6277  1 !
;  6308     6278  1 ! INPUT PARAMETERS:
;  6309     6279  1 !       P_PROMPT        - Prompt type flags:
;  6310     6280  1 !                                 Bit 0:  Prompting is enabled.
;  6311     6281  1 !                                 Bit 1:  Delete is an option.
;  6312     6282  1 !                         Default is %B'01'.
;  6313     6283  1 !
;  6314     6284  1 ! IMPLICIT INPUTS:
;  6315     6285  1 !       NONE
;  6316     6286  1 !
;  6317     6287  1 ! OUTPUT PARAMETERS:
;  6318     6288  1 !       NONE
;  6319     6289  1 !
;  6320     6290  1 ! IMPLICIT OUTPUTS:
;  6321     6291  1 !       NONE
;  6322     6292  1 !
;  6323     6293  1 ! ROUTINE VALUE:
;  6324     6294  1 !       Bit 0 set if the repair is selected, clear otherwise.
;  6325     6295  1 !       Bit 1 set if delete is selected.
;  6326     6296  1 !
;  6327     6297  1 ! SIDE EFFECTS:
;  6328     6298  1 !       NONE
;  6329     6299  1 !
;  6330     6300  1 !--
;  6331     6301  1
;  6332     6302  2 BEGIN
;  6333     6303  2 LOCAL
;  6334     6304  2         PROMPT;                                 ! Value of prompt bits
;  6335     6305  2 BUILTIN
;  6336     6306  2         ACTUALCOUNT;
;  6337     6307  2
;  6338     6308  2
;  6339     6309  2 ! If /REPAIR is not in effect, return no repair.
;  6340     6310  2 !
;  6341     6311  2 IF NOT .QUAL[QUAL_REPA]
;  6342     6312  2 THEN
;  6343     6313  2     RETURN 0;
;  6344     6314  2
;  6345     6315  2
;  6346     6316  2 ! Get PROMPT value.
;  6347     6317  2 !
;  6348     6318  2 PROMPT = %B'01';
;  6349     6319  2 IF ACTUALCOUNT() NEQ 0 THEN PROMPT = .P_PROMPT;
;  6350     6320  2
;  6351     6321  2
;  6352     6322  2 ! If /CONFIRM has been requested, do it.
;  6353     6323  2 !
;  6354     6324  2 IF .QUAL[QUAL_CONF] AND .PROMPT
;  6355     6325  2 THEN
;  6356     6326  3     BEGIN
```

```
;  6357     6327  3     LOCAL
;  6358     6328  3         ANS_BUFFER:     VECTOR[8,BYTE],         ! Buffer for response
;  6359     6329  3         ANS_DESC:       BBLOCK[8];              ! Descriptor for buffer
;  6360     6330  3
;  6361     6331  3
;  6362     6332  3     ! Set up a descriptor for the result area.
;  6363     6333  3     !
;  6364     6334  3     ANS_DESC[DSC$W_LENGTH] = 8;
;  6365     6335  3     ANS_DESC[DSC$B_DTYPE] = DSC$K_DTYPE_T;
;  6366     6336  3     ANS_DESC[DSC$B_CLASS] = DSC$K_CLASS_S;
;  6367     6337  3     ANS_DESC[DSC$A_POINTER] = ANS_BUFFER;
;  6368     6338  3
;  6369     6339  3
;  6370     6340  3     IF .PROMPT<1,1>
;  6371     6341  3     THEN
;  6372     6342  4         BEGIN
;  6373     6343  4
;  6374     6344  4         ! Prompt with delete option.
;  6375     6345  4         !
;  6376     6346  4         LIB$GET_COMMAND(
;  6377     6347  4             ANS_DESC,
;  6378     6348  4             $DESCRIPTOR('Repair this error (D to delete)? (D, Y or N): '));
;  6379     6349  4         ANS_BUFFER[0] = .ANS_BUFFER[0] AND NOT %O'040'; ! Upcase
;  6380     6350  4         IF .ANS_BUFFER[0] EQL %C'D' THEN RETURN %B'11';
;  6381     6351  4         END
;  6382     6352  3     ELSE
;  6383     6353  4         BEGIN
;  6384     6354  4
;  6385     6355  4         ! Prompt without delete option.
;  6386     6356  4         !
;  6387     6357  4         LIB$GET_COMMAND(
;  6388     6358  4             ANS_DESC,
;  6389     6359  4             $DESCRIPTOR('Repair this error? (Y or N): '));
;  6390     6360  3         END;
;  6391     6361  3     ANS_BUFFER[0] = .ANS_BUFFER[0] AND NOT %O'040';      ! Upcase
;  6392     6362  3     IF .ANS_BUFFER[0] NEQ %C'Y' THEN RETURN 0;
;  6393     6363  2     END;
;  6394     6364  2
;  6395     6365  2
;  6396     6366  2 ! Ordinary repair has been requested.
;  6397     6367  2 !
;  6398     6368  2 %B'01'
;  6399     6369  1 END;


72 72 65 20 73 69 68 74 20 72 69 61 70 65 52  0494A P.ABW:  .ASCII  \Repair this error (D to delete)? (D, Y o\ ;
65 74 65 6C 65 64 20 6F 74 20 44 28 20 72 6F  04959                                                        ;
                  6F 20 59 20 2C 44 28 20 3F 29  04968                                                        ;
                              20 3A 29 4E 20 72  04972        .ASCII  \r N): \                            ;
                                          0000002E  04978 P.ABV:  .LONG   46                              ;
                                          00000000' 0497C         .ADDRESS P.ABW                          ;
72 72 65 20 73 69 68 74 20 72 69 61 70 65 52  04980 P.ABY:  .ASCII  \Repair this error? (Y or N): \      ;
   20 3A 29 4E 20 72 6F 20 59 28 20 3F 72 6F  0498F                                                        ;
                                              0499D        .BLKB   3                                      ;
                                          0000001D  049A0 P.ABX:  .LONG   29                              ;
                                          00000000' 049A4         .ADDRESS P.ABY                          ;
```

```
                                000C 00000 DO_REPAIR:
                                           .WORD   Save R2,R3                      ; 6270
                    53 00000000'   EF  9E 00002    MOVAB   QUAL, R3                ;
                    52 00000000G   00  9E 00009    MOVAB   LIB$GET_COMMAND, R2     ;
                    5E             10  C2 00010    SUBL2   #16, SP                 ;
                4D  63             03  E1 00013    BBC     #3, QUAL, 4$            ; 6311
                    50             01  D0 00017    MOVL    #1, PROMPT              ; 6318
                                   6C  95 0001A    TSTB    (AP)                    ; 6319
                                   04  13 0001C    BEQL    1$                      ;
                    50       04    AC  D0 0001E    MOVL    P_PROMPT, PROMPT        ;
                    42             63  E9 00022 1$: BLBC    QUAL, 5$                ; 6324
                    3F             50  E9 00025    BLBC    PROMPT, 5$              ;
                    6E 010E0008    8F  D0 00028    MOVL    #17694728, ANS_DESC     ; 6334
              04    AE       08    AE  9E 0002F    MOVAB   ANS_BUFFER, ANS_DESC+4  ; 6337
                18  50             01  E1 00034    BBC     #1, PROMPT, 2$          ; 6340
                                95  AF  9F 00038    PUSHAB  P.ABV                   ; 6348
                                04  AE  9F 0003B    PUSHAB  ANS_DESC                ; 6346
                    62             02  FB 0003E    CALLS   #2, LIB$GET_COMMAND     ;
              08    AE             20  8A 00041    BICB2   #32, ANS_BUFFER         ; 6349
              44    8F       08    AE  91 00045    CMPB    ANS_BUFFER, #68         ; 6350
                                   0D  12 0004A    BNEQ    3$                      ;
                    50             03  D0 0004C    MOVL    #3, R0                  ;
                                   04  00004F    RET                             ;
                                A5 AF  9F 00050 2$: PUSHAB  P.ABX                   ; 6359
                                04 AE  9F 00053    PUSHAB  ANS_DESC                ; 6357
                    62             02  FB 00056    CALLS   #2, LIB$GET_COMMAND     ;
              08    AE             20  8A 00059 3$: BICB2   #32, ANS_BUFFER         ; 6361
              59    8F       08    AE  91 0005D    CMPB    ANS_BUFFER, #89         ; 6362
                                   03  13 00062    BEQL    5$                      ;
                                   50  D4 00064 4$: CLRL    R0                      ;
                                   04  00066    RET                             ;
                          50       01  D0 00067 5$: MOVL    #1, R0                  ; 6369
                                   04  0006A    RET                             ;

; Routine Size:  107 bytes,    Routine Base:  CODE + 49A8
```

```
;  6401     6370  1 ROUTINE EXIT_HANDLER: NOVALUE=
;  6402     6371  1
;  6403     6372  1 !++
;  6404     6373  1 !
;  6405     6374  1 ! FUNCTIONAL DESCRIPTION:
;  6406     6375  1 !       This routine is the exit handler.  It receives control when the image
;  6407     6376  1 !       is exited.  Its purpose is to unlock the volume set if necessary, and
;  6408     6377  1 !       to disable quota processing if necessary.
;  6409     6378  1 !
;  6410     6379  1 ! INPUT PARAMETERS:
;  6411     6380  1 !       Exit handler parameter list (not used).
;  6412     6381  1 !
;  6413     6382  1 ! IMPLICIT INPUTS:
;  6414     6383  1 !       NONE
;  6415     6384  1 !
;  6416     6385  1 ! OUTPUT PARAMETERS:
;  6417     6386  1 !       NONE
;  6418     6387  1 !
;  6419     6388  1 ! IMPLICIT OUTPUTS:
;  6420     6389  1 !       NONE
;  6421     6390  1 !
;  6422     6391  1 ! ROUTINE VALUE:
;  6423     6392  1 !       NONE
;  6424     6393  1 !
;  6425     6394  1 ! SIDE EFFECTS:
;  6426     6395  1 !       Volume set unlocked, if possible.  Quota processing disabled if it was
;  6427     6396  1 !       disabled on entry, if possible.
;  6428     6397  1 !
;  6429     6398  1 !--
;  6430     6399  1
;  6431     6400  2 BEGIN
;  6432     6401  2 IF .QUAL[QUAL_REPA]
;  6433     6402  2 THEN
;  6434     6403  3     BEGIN
;  6435     6404  3     CH$FILL(0, FIB$C_LENGTH, FIB);
;  6436     6405  3     FIB[FIB$W_CNTRLFUNC] = FIB$C_UNLK_VOL;
;  6437   P 6406  3     $QIOW(
;  6438   P 6407  3         FUNC=IO$_ACPCONTROL,
;  6439   P 6408  3         CHAN=.CHANNEL,
;  6440     6409  3         P1=FIB_DESC);
;  6441     6410  3     QUAL[QUAL_REPA] = 0;
;  6442     6411  2     END;
;  6443     6412  2
;  6444     6413  2
;  6445     6414  2 IF .QUOTA_DISABLE
;  6446     6415  2 THEN
;  6447     6416  3     BEGIN
;  6448     6417  3     CH$FILL(0, FIB$C_LENGTH, FIB);
;  6449     6418  3     FIB[FIB$W_CNTRLFUNC] = FIB$C_DSA_QUOTA;
;  6450   P 6419  3     $QIOW(
;  6451   P 6420  3         FUNC=IO$_ACPCONTROL,
;  6452   P 6421  3         CHAN=.CHANNEL,
;  6453     6422  3         P1=FIB_DESC);
;  6454     6423  3     QUOTA_DISABLE = 0;
;  6455     6424  2     END;
;  6456     6425  2 !
;  6457     6426  2 ! Finally, re-enable CLI CTRL/Y handling
```

```
;  6458          6427  2 !
;  6459          6428  2 LIB$ENABLE_CTRL(OLD_CTRL_MASK);
;  6460          6429  1 END;


                                                    00FC 00000 EXIT_HANDLER:
                                                               .WORD   Save R2,R3,R4,R5,R6,R7                  ;  6370
                                    57 00000000G    00   9E 00002  MOVAB   SYS$QIOW, R7                        ;
                                    56 00000000'    EF   9E 00009  MOVAB   QUAL, R6                           ;
                                 2E                 66   03 E1 00010  BBC     #3, QUAL, 1$                    ;  6401
   0040   8F                    00                 6E   00 2C 00014  MOVC5   #0, (SP), #0, #64, FIB          ;  6404
                                                       00000000' EF   0001B
                                       00000000'    EF   08 B0 00020  MOVW    #8, FIB+22                      ;  6405
                                                       7E   7C 00027  CLRQ    -(SP)                           ;  6409
                                                       7E   7C 00029  CLRQ    -(SP)                           ;
                                                       7E   D4 0002B  CLRL    -(SP)                           ;
                                             B5EC      CF   9F 0002D  PUSHAB  FIB_DESC                        ;
                                                       7E   7C 00031  CLRQ    -(SP)                           ;
                                    7E                 38   7D 00033  MOVQ    #56, -(SP)                      ;
                                             063C      C6   DD 00036  PUSHL   CHANNEL                         ;
                                                       7E   D4 0003A  CLRL    -(SP)                           ;
                                    67                 0C   FB 0003C  CALLS   #12, SYS$QIOW                   ;
                                    66                 08   8A 0003F  BICB2   #8, QUAL                        ;  6410
                                    2F          067C   C6   E9 00042 1$:  BLBC    QUOTA_DISABLE, 2$           ;  6414
   0040   8F                    00                 6E   00 2C 00047  MOVC5   #0, (SP), #0, #64, FIB          ;  6417
                                                       00000000' EF   0004E
                                       00000000'    EF   0A B0 00053  MOVW    #10, FIB+22                     ;  6418
                                                       7E   7C 0005A  CLRQ    -(SP)                           ;  6422
                                                       7E   7C 0005C  CLRQ    -(SP)                           ;
                                                       7E   D4 0005E  CLRL    -(SP)                           ;
                                             B5B9      CF   9F 00060  PUSHAB  FIB_DESC                        ;
                                                       7E   7C 00064  CLRQ    -(SP)                           ;
                                    7E                 38   7D 00066  MOVQ    #56, -(SP)                      ;
                                             063C      C6   DD 00069  PUSHL   CHANNEL                         ;
                                                       7E   D4 0006D  CLRL    -(SP)                           ;
                                    67                 0C   FB 0006F  CALLS   #12, SYS$QIOW                   ;
                                             067C      C6   D4 00072  CLRL    QUOTA_DISABLE                   ;  6423
                                             0658      C6   9F 00076 2$:  PUSHAB  OLD_CTRL_MASK               ;  6428
                                    00000000G   00      01   FB 0007A  CALLS   #1, LIB$ENABLE_CTRL           ;
                                                       04 00081  RET                                          ;  6429

; Routine Size:  130 bytes,    Routine Base:  CODE + 4A13
```

```
;  6462          6430  1 ROUTINE CTRL_AST: NOVALUE =
;  6463          6431  1
;  6464          6432  1 !++
;  6465          6433  1 !
;  6466          6434  1 ! FUNCTIONAL DESCRIPTION:
;  6467          6435  1 !       Shut down the image, a CTRL/C or CTRL/Y was entered.
;  6468          6436  1 !
;  6469          6437  1 ! INPUT PARAMETERS:
;  6470          6438  1 !       NONE
;  6471          6439  1 !
;  6472          6440  1 ! IMPLICIT INPUTS:
;  6473          6441  1 !       NONE
;  6474          6442  1 !
;  6475          6443  1 ! OUTPUT PARAMETERS:
;  6476          6444  1 !       NONE
;  6477          6445  1 !
;  6478          6446  1 ! IMPLICIT OUTPUTS:
;  6479          6447  1 !       NONE - Image exits.
;  6480          6448  1 !
;  6481          6449  1 ! ROUTINE VALUE:
;  6482          6450  1 !       NONE
;  6483          6451  1 !
;  6484          6452  1 ! SIDE EFFECTS:
;  6485          6453  1 !       NONE
;  6486          6454  1 !
;  6487          6455  1 !--
;  6488          6456  2 BEGIN
;  6489          6457  2 !
;  6490          6458  2 ! Re-enable CTRL/C and CTRL/Y in case user types another while we
;  6491          6459  2 ! are shutting down.
;  6492          6460  2 !
;  6493        P 6461  2 $QIOW(FUNC=(IO$_SETMODE OR IO$M_CTRLCAST),
;  6494        P 6462  2       CHAN=.CHANNEL_TT,
;  6495        P 6463  2       IOSB=IOSB,
;  6496        P 6464  2       P1=CTRL_AST,
;  6497        P 6465  2       P2=0,
;  6498          6466  2       P3=PSL$C_USER);
;  6499        P 6467  2 $QIOW(FUNC=(IO$_SETMODE OR IO$M_CTRLYAST),
;  6500        P 6468  2       CHAN=.CHANNEL_TT,
;  6501        P 6469  2       IOSB=IOSB,
;  6502        P 6470  2       P1=CTRL_AST,
;  6503        P 6471  2       P2=0,
;  6504          6472  2       P3=PSL$C_USER);
;  6505          6473  2 !
;  6506          6474  2 ! Call the exit handler now, before we re-enable cli ctrl handling
;  6507          6475  2 !
;  6508          6476  2 SIGNAL(VERIFY$_ABORT);
;  6509          6477  2 EXIT_HANDLER();
;  6510          6478  2 $EXIT(CODE=SS$_NORMAL);
;  6511          6479  1 END;


                                                .EXTRN  SYS$EXIT

                                000C 00000 CTRL_AST:
                                                .WORD   Save R2,R3                      ; 6430
```

```
                        53 00000000G   00   9E 00002     MOVAB   SYS$QIOW, R3                                ;
                        52 00000000'   EF   9E 00009     MOVAB   IOSB, R2                                    ;
                                       7E   7C 00010     CLRQ    -(SP)                                       ; 6466
                                    7E 03   7D 00012     MOVQ    #3, -(SP)                                   ;
                                       7E   D4 00015     CLRL    -(SP)                                       ;
                                       E6   AF 9F 00017  PUSHAB  CTRL_AST                                    ;
                                       7E   7C 0001A     CLRQ    -(SP)                                       ;
                                       52   DD 0001C     PUSHL   R2                                          ;
                                 7E  0123   8F   3C 0001E  MOVZWL  #291, -(SP)                               ;
                                00000000'   EF   DD 00023  PUSHL   CHANNEL_TT                                ;
                                       7E   D4 00029     CLRL    -(SP)                                       ;
                                    63 0C   FB 0002B     CALLS   #12, SYS$QIOW                               ;
                                       7E   7C 0002E     CLRQ    -(SP)                                       ; 6472
                                    7E 03   7D 00030     MOVQ    #3, -(SP)                                   ;
                                       7E   D4 00033     CLRL    -(SP)                                       ;
                                       C8   AF 9F 00035  PUSHAB  CTRL_AST                                    ;
                                       7E   7C 00038     CLRQ    -(SP)                                       ;
                                       52   DD 0003A     PUSHL   R2                                          ;
                                 7E  A3   8F   9A 0003C  MOVZBL  #163, -(SP)                                 ;
                                00000000'   EF   DD 00040  PUSHL   CHANNEL_TT                                ;
                                       7E   D4 00046     CLRL    -(SP)                                       ;
                                    63 0C   FB 00048     CALLS   #12, SYS$QIOW                               ;
                                00000000G   8F   DD 0004B  PUSHL   #VERIFY$_ABORT                             ; 6476
                    00000000G   00          01   FB 00051  CALLS   #1, LIB$SIGNAL                           ;
                        FF21    CF          00   FB 00058  CALLS   #0, EXIT_HANDLER                         ; 6477
                                            01   DD 0005D  PUSHL   #1                                        ; 6478
                    00000000G   00          01   FB 0005F  CALLS   #1, SYS$EXIT                             ;
                                            04 00066     RET                                                 ; 6479

; Routine Size:  103 bytes,    Routine Base:  CODE + 4A95
```

```
;  6513     6480  1 ROUTINE CHECK_DATE(ODS2_DATE,ODS1_DATE,MESSAGE,FILE_ID,HEADER): NOVALUE=
;  6514     6481  1
;  6515     6482  1 !++
;  6516     6483  1 !
;  6517     6484  1 ! FUNCTIONAL DESCRIPTION:
;  6518     6485  1 !       This routine checks and corrects a date field.
;  6519     6486  1 !
;  6520     6487  1 ! INPUT PARAMETERS:
;  6521     6488  1 !       ODS2_DATE       - Pointer to ODS-2 date within header.
;  6522     6489  1 !       ODS1_DATE       - Pointer to ODS-1 date within header.
;  6523     6490  1 !       MESSAGE         - Message to be issued.
;  6524     6491  1 !       FILE_ID         - File identification.
;  6525     6492  1 !       HEADER          - Pointer to header.
;  6526     6493  1 !
;  6527     6494  1 ! IMPLICIT INPUTS:
;  6528     6495  1 !       NONE
;  6529     6496  1 !
;  6530     6497  1 ! OUTPUT PARAMETERS:
;  6531     6498  1 !       NONE
;  6532     6499  1 !
;  6533     6500  1 ! IMPLICIT OUTPUTS:
;  6534     6501  1 !       NONE
;  6535     6502  1 !
;  6536     6503  1 ! ROUTINE VALUE:
;  6537     6504  1 !       NONE
;  6538     6505  1 !
;  6539     6506  1 ! SIDE EFFECTS:
;  6540     6507  1 !       NONE
;  6541     6508  1 !
;  6542     6509  1 !--
;  6543     6510  1
;  6544     6511  2 BEGIN
;  6545     6512  2 MAP
;  6546     6513  2         ODS2_DATE:      REF VECTOR,             ! Pointer to quadword date
;  6547     6514  2         ODS1_DATE:      REF BBLOCK,             ! Pointer to ODS-1 date
;  6548     6515  2         FILE_ID:        REF BBLOCK;             ! Pointer to file ID
;  6549     6516  2 LITERAL
;  6550     6517  2         DATE_LENGTH=    23;
;  6551     6518  2 LOCAL
;  6552     6519  2         DATE_BUFFER:    BBLOCK[DATE_LENGTH],    ! Buffer for $BINTIM
;  6553     6520  2         DATE_TEMP:      VECTOR[2],              ! Binary date value
;  6554     6521  2         DATE:           REF VECTOR,             ! Pointer to comparison date
;  6555     6522  2         DESC:           VECTOR[2];              ! Descriptor for buffer
;  6556     6523  2
;  6557     6524  2
;  6558     6525  2 ! Assume that the comparison date is the ODS-2 date.
;  6559     6526  2 !
;  6560     6527  2 DATE = .ODS2_DATE;
;  6561     6528  2
;  6562     6529  2
;  6563     6530  2 ! For ODS-1, convert the date to 64-bit format.
;  6564     6531  2 !
;  6565     6532  2 IF .STRUCTURE_LEVEL EQL 1
;  6566     6533  2 THEN
;  6567     6534  3     BEGIN
;  6568     6535  3
;  6569     6536  3     ! The comparison date is the temporary buffer.
```

```
;  6570     6537  3      !
;  6571     6538  3      DATE = DATE_TEMP;
;  6572     6539  3
;  6573     6540  3
;  6574     6541  3      IF .ODS1_DATE[0,0,8,0] EQL 0
;  6575     6542  3      THEN
;  6576     6543  4          BEGIN
;  6577     6544  4          !
;  6578     6545  4          ! Null date.  If the date is all nulls, set it to zeros so that it
;  6579     6546  4          ! will surely pass the comparison.  If it is not, set it to ones so
;  6580     6547  4          ! that it will surely fail.
;  6581     6548  4          !
;  6582     6549  4          DATE_TEMP[0] = DATE_TEMP[1] = 0;
;  6583     6550  4          IF CRSFIND_NOT_CH(13, .ODS1_DATE, 0) NEQ 0
;  6584     6551  4              THEN DATE_TEMP[0] = DATE_TEMP[1] = -1;
;  6585     6552  4          END
;  6586     6553  3      ELSE
;  6587     6554  4          BEGIN
;  6588     6555  4          !
;  6589     6556  4          ! Convert the date to a format acceptable to $BINTIM.
;  6590     6557  4          !
;  6591     6558  4          DATE_BUFFER[0,0,16,0] = .ODS1_DATE[0,0,16,0];
;  6592     6559  4          DATE_BUFFER[2,0,8,0] = '-';
;  6593     6560  4          DATE_BUFFER[3,0,24,0] = .ODS1_DATE[2,0,24,0];
;  6594     6561  4          DATE_BUFFER[6,0,24,0] = '-19';
;  6595     6562  4          DATE_BUFFER[9,0,16,0] = .ODS1_DATE[5,0,16,0];
;  6596     6563  4          DATE_BUFFER[11,0,8,0] = ' ';
;  6597     6564  4          DATE_BUFFER[12,0,16,0] = .ODS1_DATE[7,0,16,0];
;  6598     6565  4          DATE_BUFFER[14,0,8,0] = ':';
;  6599     6566  4          DATE_BUFFER[15,0,16,0] = .ODS1_DATE[9,0,16,0];
;  6600     6567  4          DATE_BUFFER[17,0,8,0] = ':';
;  6601     6568  4          DATE_BUFFER[18,0,16,0] = .ODS1_DATE[11,0,16,0];
;  6602     6569  4          DATE_BUFFER[20,0,24,0] = '.00';
;  6603     6570  4
;  6604     6571  4
;  6605     6572  4          ! Try to convert the date using $BINTIM.  If this fails, set the date
;  6606     6573  4          ! to all ones so that it will surely fail the comparison.
;  6607     6574  4          !
;  6608     6575  4          DESC[0] = DATE_LENGTH;
;  6609     6576  4          DESC[1] = DATE_BUFFER;
;  6610     6577  5          IF NOT $BINTIM(TIMBUF=DESC, TIMADR=DATE_TEMP)
;  6611     6578  4          THEN
;  6612     6579  4              DATE_TEMP[0] = DATE_TEMP[1] = -1;
;  6613     6580  3          END;
;  6614     6581  2      END;
;  6615     6582  2
;  6616     6583  2
;  6617     6584  2  ! Compare date in quadword format.  If the comparison fails, issue the message,
;  6618     6585  2  ! reconstruct the date as appropriate, and rewrite the header.  Avoid this for
;  6619     6586  2  ! the revision date of the index file.  Because we are accessing and
;  6620     6587  2  ! deaccessing index files on the second channel, the revision date may properly
;  6621     6588  2  ! become later than the value we have for the current time.
;  6622     6589  2  !
;  6623     6590  2  IF
;  6624     6591  2      .DATE[1] GTRU .CURRENT_TIME[1] OR
;  6625     6592  2      .DATE[1] EQL .CURRENT_TIME[1] AND .DATE[0] GTRU .CURRENT_TIME[0]
;  6626     6593  2  THEN
```

```
;  6627          6594  3     BEGIN
;  6628          6595  3     IF
;  6629          6596  3         .FILE_ID[FID$W_NUM] NEQ FID$C_INDEXF OR .FILE_ID[FID$B_NMX] NEQ 0 OR
;  6630          6597  3         .MESSAGE NEQ VERIFY$_FUTREVDAT
;  6631          6598  3     THEN
;  6632          6599  4         BEGIN
;  6633          6600  4         HEADER_ERROR(,MESSAGE, .FILE_ID, .HEADER);
;  6634          6601  4         IF DO_REPAIR()
;  6635          6602  4         THEN
;  6636          6603  5             BEGIN
;  6637          6604  5             IF .STRUCTURE_LEVEL EQL 2
;  6638          6605  5             THEN
;  6639          6606  6                 BEGIN
;  6640          6607  6                 ODS2_DATE[0] = .CURRENT_TIME[0];
;  6641          6608  6                 ODS2_DATE[1] = .CURRENT_TIME[1];
;  6642          6609  6                 END
;  6643          6610  5             ELSE
;  6644          6611  5                 CH$MOVE(13, CURRENT_TIME_1, .ODS1_DATE);
;  6645          6612  5             WRITE_HEADER(.FILE_ID, .HEADER);
;  6646          6613  5             END;
;  6647          6614  4         END;
;  6648          6615  2     END;
;  6649          6616  1 END;


                                                                .EXTRN  SYS$BINTIM

                                                00FC 00000 CHECK_DATE:
                                                                .WORD   Save R2,R3,R4,R5,R6,R7                  ; 6480
                                    57 00000000'    EF  9E 00002        MOVAB   STRUCTURE_LEVEL, R7
                                    5E              28  C2 00009        SUBL2   #40, SP
                                    54          04  AC  D0 0000C        MOVL    ODS2_DATE, R4                  ; 6527
                                    53              54  D0 00010        MOVL    R4, DATE
                                    01              67  D1 00013        CMPL    STRUCTURE_LEVEL, #1            ; 6532
                                                    03  13 00016        BEQL    1$
                                                0080    31 00018        BRW     5$
                                    53          08  AE  9E 0001B 1$:    MOVAB   DATE_TEMP, DATE                ; 6538
                                    52          08  AC  D0 0001F        MOVL    ODS1_DATE, R2                  ; 6541
                                                62  95 00023        TSTB    (R2)
                                                11  12 00025        BNEQ    3$
                                                08  AE  7C 00027        CLRQ    DATE_TEMP                      ; 6549
                                62              0D      00  3B 0002A        SKPC    #0, #13, (R2)              ; 6550
                                                02  12 0002E        BNEQ    2$
                                                51  D4 00030        CLRL    R1
                                                51  D5 00032 2$:     TSTL    R1
                                                65  13 00034        BEQL    5$
                                                5B  11 00036        BRB     4$                             ; 6551
                                        10  AE          62  B0 00038 3$:     MOVW    (R2), DATE_BUFFER          ; 6558
                                        12  AE          2D  90 0003C        MOVB    #45, DATE_BUFFER+2         ; 6559
13  AE                  18              00      02  A2  F0 00040        INSV    2(R2), #0, #24, DATE_BUFFER+3  ; 6560
16  AE                  18              00 00393120 8F  F0 00047        INSV    #3748141, #0, #24, DATE_BUFFER+6 ; 6561
                                    19  AE          05  A2  B0 00051        MOVW    5(R2), DATE_BUFFER+9       ; 6562
                                    1B  AE          20  90 00056        MOVB    #32, DATE_BUFFER+11        ; 6563
                                    1C  AE          07  A2  B0 0005A        MOVW    7(R2), DATE_BUFFER+12      ; 6564
                                    1E  AE          3A  90 0005F        MOVB    #58, DATE_BUFFER+14        ; 6565
                                    1F  AE          09  A2  B0 00063        MOVW    9(R2), DATE_BUFFER+15      ; 6566
```

```
                   21  AE              3A  90 00068        MOVB    #58, DATE_BUFFER+17                        ; 6567
                   22  AE          0B  A2  B0 0006C        MOVW    11(R2), DATE_BUFFER+18                     ; 6568
   24  AE      18  00  0030302E        8F  F0 00071        INSV    #3158062, #0, #24, DATE_BUFFER+20          ; 6569
                       6E              17  D0 0007B        MOVL    #23, DESC                                  ; 6575
                   04  AE          10  AE  9E 0007E        MOVAB   DATE_BUFFER, DESC+4                        ; 6576
                                   08  AE  9F 00083        PUSHAB  DATE_TEMP                                  ; 6577
                                   04  AE  9F 00086        PUSHAB  DESC                                       ;
            00000000G  00              02  FB 00089        CALLS   #2, SYS$BINTIM                             ;
                       08              50  E8 00090        BLBS    R0, 5$                                     ;
                   0C  AE              01  CE 00093 4$:    MNEGL   #1, DATE_TEMP+4                            ; 6579
                   08  AE              01  CE 00097        MNEGL   #1, DATE_TEMP                              ;
                   14  A7          04  A3  D1 0009B 5$:    CMPL    4(DATE), CURRENT_TIME+4                    ; 6591
                                       08  1A 000A0        BGTRU   6$                                         ;
                                       4E  12 000A2        BNEQ    10$                                        ; 6592
                   10  A7              63  D1 000A4        CMPL    (DATE), CURRENT_TIME                       ;
                                       48  1B 000A8        BLEQU   10$                                        ;
                       56          10  AC  D0 000AA 6$:    MOVL    FILE_ID, R6                                ; 6596
                       01              66  B1 000AE        CMPW    (R6), #1                                   ;
                                       0F  12 000B1        BNEQ    7$                                         ;
                                   05  A6  95 000B3        TSTB    5(R6)                                      ;
                                       0A  12 000B6        BNEQ    7$                                         ;
            00000000G  8F          0C  AC  D1 000B8        CMPL    MESSAGE, #VERIFY$_FUTREVDAT                ; 6597
                                       30  13 000C0        BEQL    10$                                        ;
                                   14  AC  DD 000C2 7$:    PUSHL   HEADER                                     ; 6600
                                       56  DD 000C5        PUSHL   R6                                         ;
                                   0C  AC  DD 000C7        PUSHL   MESSAGE                                    ;
                 EE55  CF              03  FB 000CA        CALLS   #3, HEADER_ERROR                           ;
                 FDD8  CF              00  FB 000CF        CALLS   #0, DO_REPAIR                              ; 6601
                       1B              50  E9 000D4        BLBC    R0, 10$                                    ;
                       02              67  D1 000D7        CMPL    STRUCTURE_LEVEL, #2                        ; 6604
                                       06  12 000DA        BNEQ    8$                                         ;
                       64          10  A7  7D 000DC        MOVQ    CURRENT_TIME, (R4)                         ; 6607
                                       06  11 000E0        BRB     9$                                         ; 6604
          08  BC   18  A7              0D  28 000E2 8$:    MOVC3   #13, CURRENT_TIME_1, @ODS1_DATE            ; 6611
                                   14  AC  DD 000E8 9$:    PUSHL   HEADER                                     ; 6612
                                       56  DD 000EB        PUSHL   R6                                         ;
                 EADE  CF              02  FB 000ED        CALLS   #2, WRITE_HEADER                           ;
                                           04 000F2 10$:   RET                                                ; 6616

; Routine Size:  243 bytes,    Routine Base:  CODE + 4AFC
```

```
; 6651          6617  1 END
; 6652          6618  0 ELUDOM


                                                        .EXTRN  LIB$SIGNAL

;                               PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  DATA                          35676  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  CODE                          19439  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)


;                           Library Statistics
;
;                                    -------- Symbols --------      Pages       Processing
;       File                         Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]LIB.L32;1      18619       350         1      1000          00:01.9


; Information:  1
; Warnings:     0
; Errors:       0


;                                       COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:VERIFY/OBJ=OBJ$:VERIFY MSRC$:VERIFY/UPDATE=(ENH$:VERIFY)

; Size:          18284 code + 36831 data bytes
; Run Time:        04:47.1
; Elapsed Time:    09:30.7
; Lines/CPU Min:    1382
; Lexemes/CPU-Min: 21775
; Memory Used:  1160 pages
; Compilation Complete
```

0431 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

WELCOME
TXT

LOGIN
COM

SYSTARTUP
COM

XQBOOT
LIS

EDTINI
EDT

REMOVE
COM

VERIFY
LIS

VERIFY

UVMS

UVSTARTUP
COM

VERIFY
MAP

ADDUSER
COM

MODPARAMS
DAT

RESTUSER
COM

BACKUSER
COM

SUCCESS
TXT

SYLOGIN
COM

SCRSHR
MAP

DSTDEF
SDL

F11A
SDL

VERIFYMSG
LIS

LIBICQDEF
SDL

F11DEF
SDL

ACCDEF
MDL

VMSLIB

OBJFMT
SDL